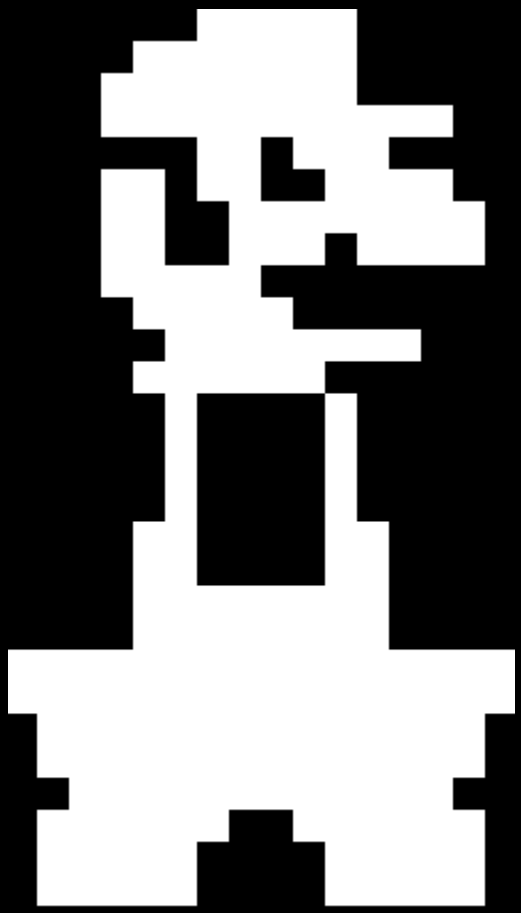
Russell's Quarterly

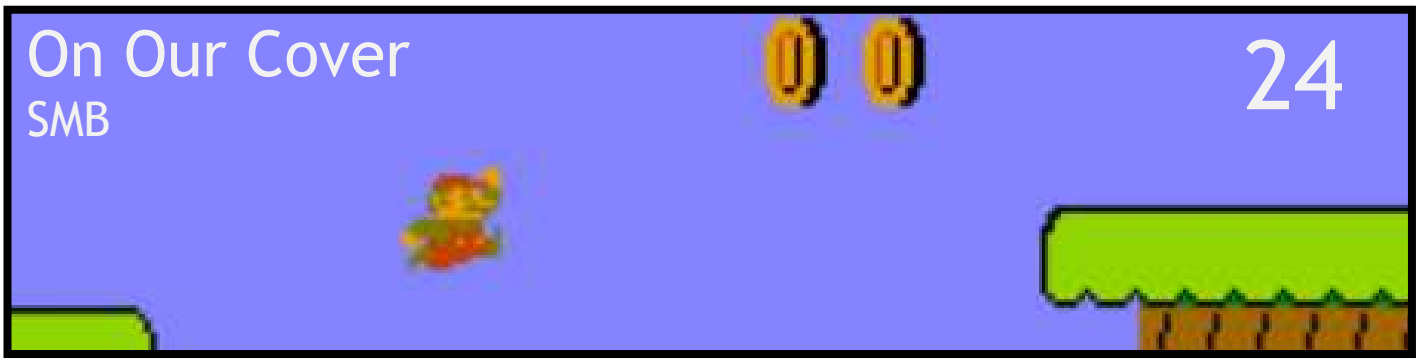
the free online magazine of video game design

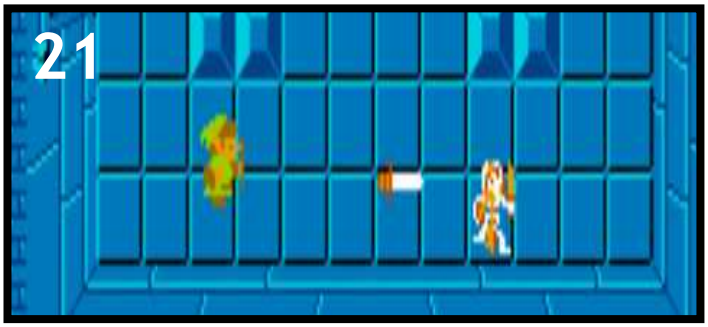


No. 5
SMB

Russell's Quarterly

NUMBER 5, EARLY WINTER 2009.





03 **Preamble**

ARTICLES

04 **Chance and Skill**
Keeping it interesting while keeping it fair.
TOM RUSSELL

13 **Sprite Substance**
Creating a visual language.
TOM RUSSELL

21 **Miyamoto's Framework**
Why it works, and a look at other possibilities.
PAUL ERES

ON OUR COVER

24 **SMB: Super Mario Bros.**
Cracking open the textbook of game design.

LOOK CLOSER

53 **The Narshe Incident**
Attack power as a source of meaning in FFVI.
TOM RUSSELL

REVIEWS

57 **Pinball Panda**

60 **Alternate Path**

SPECIAL COMMENT

63 **Gaming With One Hand**
Does the Wii discriminate?
PETER JURICH

REDESIGN BY **MARY RUSSELL**

PREAMBLE.

NEWS AND NOTES

I often draw on the work of game designer (*Kongai)* and championship *Street Fighter Turbo* player **David Sirlin**. Mr. Sirlin's amazingly informative website, **sirlin.net**, went through a major overhaul in November. This is not merely cosmetic; there is a plethora not only of new content but of revised content. Whole articles have been rethought and rewritten completely. His new version of his slippery slope article, for example, improves on his already excellent and thought-provoking original. As always, I highly recommend Mr. Sirlin's site.

In more egocentric news, the Quarterly recently won the **Game Maker Blog Community Award for Best Reviews of Game Maker Games.** We were also nominated, but did not take, the award for **Best Game Maker Magazine.** Our thanks to everyone who nominated/voted for us, and I tip my hat to **Game Maker Blog** for providing the best Game Maker-related news and commentary on the web.

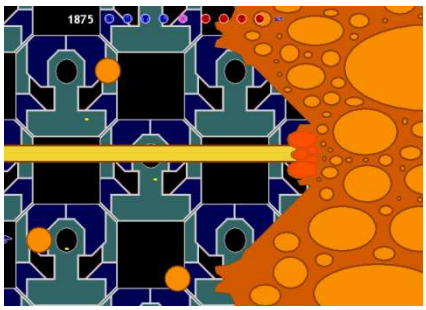
And while I'm handing out Quarterly-related kudos, I'd like to thank **EliteIndie.com** for hosting the Quarterly in its preferred PDF format, and **dopterra.com** for providing us with our own web space. That's right, the Quarterly now has an official website: http://dopterra.com/russell/home.mht. We'll be adding HTML versions of our articles as well as web-only extras slowly but surely.

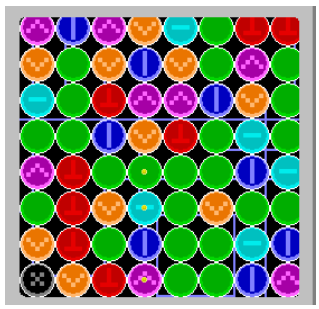
CONTRIBUTORS

Paul "Rinkuhero" Eres is the game designer and programmer who created **Immortal Defense** and *Alphasix*, among others. He's also the creator of the **Tomato Engine** that powers **Fedora Spade**. Paul wrote his thought-provoking essay "**Miyamoto's Framework**" on his blog nearly two years ago, and he was kind enough to allow us to reprint it here (page 21). You can visit Paul and find out more about his current projects, like **Saturated Dreamers,** at www.studioeres.com.

Peter Jurich is the author of **Typing With One Hand**, a harrowing but funny memoir about surviving Catholic school, his parents' divorce, a brain tumor, and adolescence. The entire book is available for free(typingwithonehand.blogspot.com) and you can see how he's doing by visiting him over at sympathyforthepencil.blogspot.com. His piece **Gaming With One Hand** reflects on the downside of the Nintendo Wii and its innovative concept.

Modesty prevents me from plugging our recently-completed film, *Son of a Seahorse*. I certainly won't be directing you to its super-cool awesome website, sonofaseahorse.blogspot.com, where you'll find all manner of screenshots and updates from The Dreaded Festival Circuit, any more than I'd point your way to my recently completed games, **Daizi** (yoyogames.com/games/show/59364) and **Eksterland** (yoyogames.com/games/show/60419).





Articles, editorials, and works of fiction, along with reviews requests and commentary, both positive and not so positive, can be submitted to the editor at **russellsquarterly@yahoo.com**. All images are copyright their respective copyright holders, and are used here for the purposes of academic criticism. **Tom Russell, Editor.**

Chance and Skill
by Tom Russell

Generally, my wife and I eat dinner in our living room, between six-thirty and seven, so that we can watch NBC's national news broadcast. Following the news on our local station are two game show programs, one of which is very good and thus the product of excellent game design and one of which is lacking on both counts.

The well-designed game show is *Jeopardy!* A quiz show, it demands and rewards skill. Not only must players have a wide-ranging pool of knowledge, but they must draw on it with lightning speed. Each round of play also features at least one pun-centered or wordplay-based category that requires actual outside-the-box thinking rather than a capacity for trivia. Contestants largely succeed and fail on their merits. I've often heard host Alex Trebek refer to a champion's "earnings" as opposed to "winnings", and I think that distinction is quite apt.

The poorly-designed game that precedes it is *Wheel of Fortune*. While there is *some* skill required in what is basically a glorified version of the old Hangman guessing game, ninety percent of the game is really luck. The money won for correctly guessing a given letter depends on a spin of the titular wheel and can range from three hundred dollars to as much as one million. There is also the dreaded bankrupt wedge, which completely clears out any contestant's money or prizes for the current round.

Think about that for a moment. You might have guessed correctly letter-after-letter, you might be on the cusp of solving the puzzle-- it's on the tip of your tongue!-- you've won more than ten thousand dollars in a single round and, boom, it lands on bankrupt. You lose all that money and the smug football jock besides you who can't even spell his own name solves the puzzle and wins a trip to the Bahamas.

Jeopardy! is a game of skill, which rewards ability, while *Wheel of Fortune* is a game of chance, which rewards nothing. It is about as exciting and interesting as a coin toss or a roll of a die. The only thing it has going for it is Vanna White, still a stunning vision of womanhood. (If instead of turning over letters she turned over clues in *Jeopardy!*, I can almost guarantee *Wheel*'s audience would evaporate overnight.)

Games of skill are superior in every way to games of chance. And yet... and yet...

The two poles in practice are not nearly so polar. Just as *Wheel* has some element of skill, *Jeopardy!* has some element of chance: the positions of the daily doubles, the selection of a categories themselves, the final clue. Chance can be utilized as a test of skill in all games-- which brings us, of course, to video games.

Routes by Rote

Though, as regular readers are no doubt aware, I am unabashed in my enthusiasm for older games, even I must admit that many of these games-- including, yes, my beloved **Super Mario Bros.**-- do not test a player's mastery of game play skills so much as they test the player's skills in specific situations. That is, rather than testing your ability to jump over pits or avoid traps, they test your ability to jump over this specific pit and to avoid that specific trap. Failing to do so does not teach you how to make that jump or avoid that trap-- it only teaches you to beware of that trap or to be ready for that jump.

Indeed, that is why older games are such a good fit for time attacks and speedruns. A successful speedrun, even one hampered by seemingly punishing criteria such as "no deaths" or "no hits", is less a matter of technical skill than it is a matter of route-planning, of paying attention to patterns. Now, I'm not saying that speedrunning doesn't require technical skill, as they very obvious do; their prowess is beyond impressive and I am constantly in awe of what they can do. But even the best of them will admit that it's really a matter of memorizing level layout and enemy attack patterns.

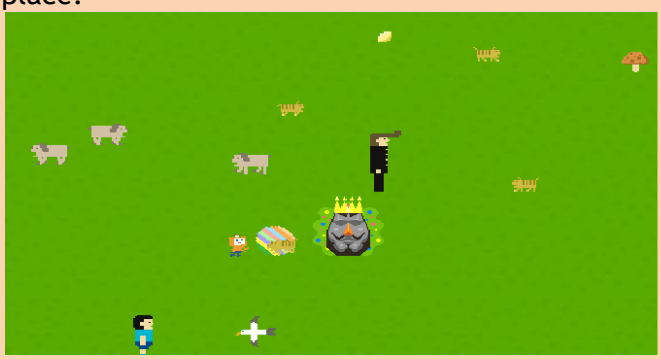
To better illustrate what I'm talking about, imagine, if you will, if *Jeopardy!* asked its contestants to answer the same questions episode after episode. The same categories, the same clues, the same daily double positions, and the same final. What was once a test of skill now becomes a test of skills in these and only these specific and immutable parameters, and thus is really no test of skill at all, save the skill of memorization. And the wordplay categories-- the ones that depend on puns, on odd associations, and on those letters in quotation marks-- which are intended as a guard against memorization by placing the emphasis on creative thinking skills as opposed to factoids would be rendered useless if they were the same night after night after night.

Route-Planning as Game Play

While **Katamari Damacy** and its sequels are the games in which "you roll stuff up in that freaky ball", a deeper look at its high concept reveals that they're really about route-planning. More-so than any other game, and more successfully than any other game, this series foregrounds the process of trial-and-error, of learning the layout of a stage, of internalizing the sizes and shapes of the various objects in relation to one another, and of forming a strategy to accomplish the level's stated goal. You'll almost never succeed your first time through a level; indeed, it'll take two or three or sometimes even eight or nine attempts to master it.

A big part of why the game is so successful and playable instead of frustrating and monotonous is that, unlike other games that depend on route-planning, there is no immediate "penalty" for going the wrong way. You can't die in the **Katamari** games, whereas if you don't know what awaits you on the next platform in **Adventure Island**, you're liable to start the level (or the game) over. Because there is no penalty, you're encouraged to explore; and since you're given a huge 3-D stage to explore, there's no one "right" way to complete a level. Here's a game, then, that uses route-planning to foreground the process of *discovery*: a powerful thing, that.

That's why the two-player mode of **Katamari Damacy**, with its limited environment, cramped split-screen, and competitive nature really works against the game. Simply rolling stuff up isn't enough to be fun; route-planning is. And that's why Namco's online flash version, which puts the player in a free-form limbo in which various objects float around you, is so terrible: it cuts out the entire point of the game in the first place.



That's one reason, indeed, why so many puzzle games, such as the much-beloved **Lolo/Eggerland** series, do not hold up nearly as well the second time around. The first time around, the puzzles tax your ability to think your way through a seemingly impossible situation. But once you've solved it, the puzzles tax and challenge you no longer; the second time through, the puzzle becomes lifeless and empty.

Platformers and other action games are less likely to become "hollowed out" by the second or even tenth play-through, even if the arrangement of their various elements remain the same. Partly, however, this is because of the sheer number of choices available. While I think *Jeopardy!* would be several measures worse if it featured the same exact categories and clues every night, by allowing the contestants in either itineration to select any one of thirty clues, it would still be measures above *Wheel of Fortune* if that game had a similar constriction. If *Wheel of Fortune* was the same every night, its linear puzzle layout would result in tedium by the second or third play-through. It would be like a block puzzle game.

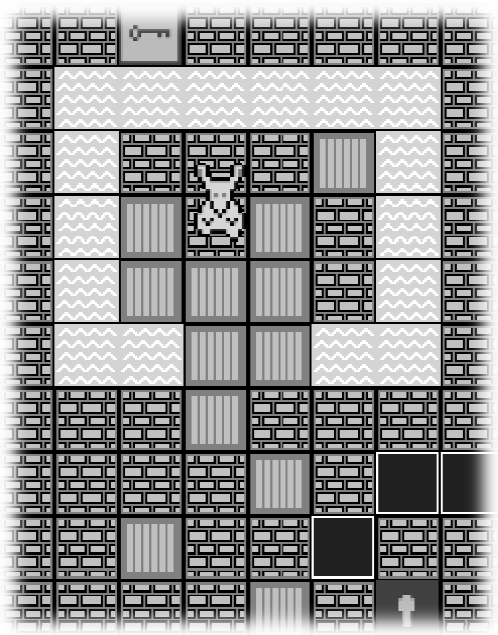
But *Jeopardy!* would be more like a platformer, in that there are a number of options available to the player. And like the platformers of old, it wouldn't merely be memorization that helps a player succeed but optimization and strategy. The Final Jeopardy wagers would not be as strategic; if everyone knows the answer, everyone is going to wager every cent. The game would be won by whoever was able to take the largest lead the quickest. Expert players would do the equivalent of a speedrunner's route-planning, grabbing the Daily Doubles before the other guy but at a moment that would still be optimal and allow them to increase their winnings. And they would still run circles around players who whine about "playing fair" and "playing for fun", refusing to memorize clue locations and exploit the system to their benefit.

At the same time: who would really *want* to have a *Jeopardy!* that played like that? While it might hold up better than *Wheel* under similar circumstances, it will, in the end, cease to be interesting and dramatic. It is the random element-- the fact that no one knows what clues or categories will be featured, where the Daily Doubles will be, what the final will be or what the other guy is going to wager-- that makes it competitive, interesting, and fun after twenty-odd years.

And, in this day and age in which players are more likely upon a game's completion to trade it in for another than to play it a second time, a game designer has to think twice about presenting the same elements in the same order moving in the same patterns over and over again.

Randomness vs. Fairness

A degree of randomness in game design, not only in enemy behaviours and artificial intelligence but also in terms of the arrangement of certain elements, can't be a bad thing. But it's not quite as simple and pat as that; injecting randomness into a winnable game creates a number of practical problems, most of which revolve around that cornerstone of player advocacy and thus game design, fairness.[1]



Imagine, for example, a randomly-generated block puzzle. What are the chances that that puzzle will have an actual solution? For that matter, if the random arrangement of blocks and hazards *did* produce a puzzle with a solution, what are the chances that that would be the only solution? The special pleasures of block puzzle games-- which depend on each puzzle having one single frustrating

[1] We're restricting our current discussion to only winnable games; Lose Better games, such as **Tetris**, often make a great deal of use of randomness but it is in those cases far easier to implement with a degree of fairness.

and yet so-obvious-why-didn't-I-think-of-it-before solution that makes use of all the tools available with no fluff or fat-- would be considerably muted by a random approach.

These are the two pitfalls of randomly-generated content: one possibility is difficult to the point of being literally impossible and one is so easy that it cannot be said to really test any kind of skills or, indeed, provide any kind of pleasure-- not the non-mimetic pleasures of game play nor the mimetic pleasures provided by the artful arrangement of visual elements on the screen (rather, it's all the more likely to get very messy and very ugly very quickly).

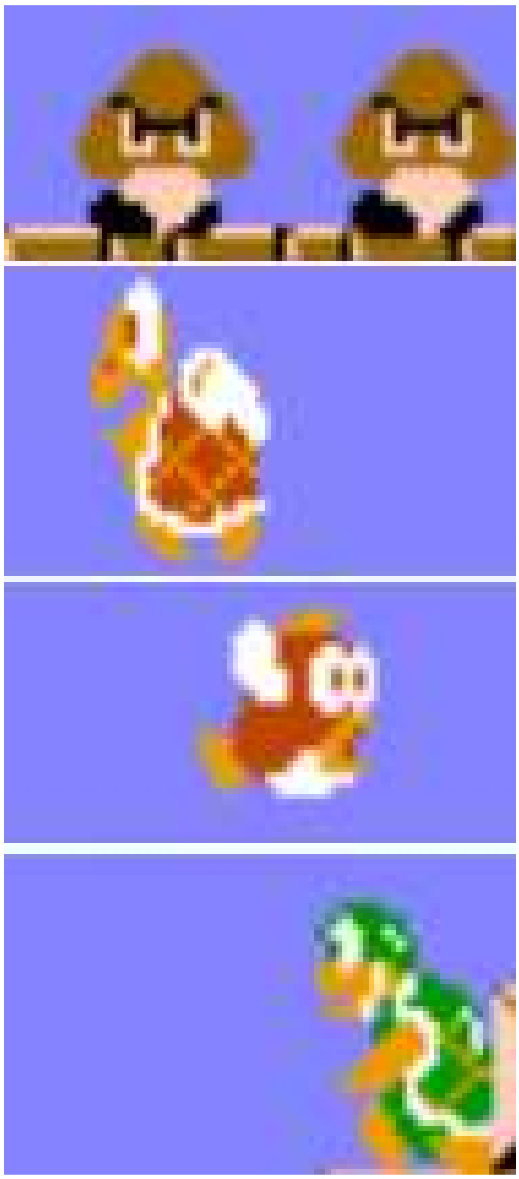
With a tip of the hat to Sister State-the-Obvious: in order for a winnable game to be winnable, there has to be a way to win it. Puzzles need solutions. Obstacles must be within the player's ability to overcome them. Invincible enemies should not be charging a defenseless player en masse and one should never have to go through ten levels without a power-up or checkpoint in sight. And this is why, of course, good game designers-- or, to be strictly accurate, good game programmers-- find ways to use randomly-created objects, circumstances, and level layouts to their advantage, minimizing potential pitfalls through a complex web of if-then statements, number sequences, variables, and algorithms.[2]

But, since we're not all masters of programming, let's look at some plain-English solutions to some of these problems by tackling the obvious example, enemies: how do we balance (1) the desire for enemies to be randomly-generated instead of appearing in the same location every time, (2) the need for that random generation of enemies to result in enemies that are tough enough to give our player a tussle, (3) the need for those enemies not to overwhelm the player by sheer numbers and (4) the need to ensure that the enemies are beatable-- i.e. that an enemy that can only be defeated by a fire weapon only appears when the player has a fire weapon?

That last point is perhaps the simplest; you can either (a) create a fire weapon variable and only allow that monster to be generated when the variable is a "yes" or a "1", or (b) not include those sort of specific weapon-to-monster relationships in your game in the first place. This brings up an important concept in game design that I'm sure we'll be returning to through-out the life of this magazine; if something's not working, don't be afraid to chuck it, genre standards be damned.

Now, let's address points two and three by creating a series of tiers. The least powerful and threatening enemies occupy the bottom tier, the most powerful and threatening enemies are in the top tier, and the other enemies are dispersed in however-many tiers in between. When we want our game to generate some baddies for our players to beat on, it won't draw from a pool of every available enemy type but rather enemies within a given tier depending on the circumstances.

(continued on page following)



Tiers; easy to hard.

[2] See our **NetHack** sidebar.

NetHack

One genre that increasingly relies on the random-generation of content is the dungeon-crawler. Each time you play a rogue-like such as **NetHack,** you are faced with a completely different fifty-level dungeon with lots of rooms, monsters, treasures, traps, and secret passages. While it's true that some paths do lead to nowhere, each level has to have an exit and there has to be a way for the player to reach or discover that exit. And, in **NetHack**, a given level's exit is always reachable.

Curious as to how a game with such complexity is able to pull this off, I went searching for the answer the only place any self-respecting person would: USENET. Posting my question on rec.games.rougelike.nethack, I received the following and immensely helpful response from one Aardvark Joe:

For what it's worth, the code to create a level is somewhat cryptic, even by Nethack standards, so it's not too surprising that you wouldn't be able to follow it if you don't understand C. I've actually never dived into it much before, but there's always a first time for everything, so I'll give it a shot. For those who are interested, this happens in src/mklev.c, function makelevel(), although I'll gloss over some stuff. Those who know the code better than me, feel free to make corrections ... I'm sure I don't have it all exactly right.

The first thing that happens is Nethack determines (based on the level being created) if this is a special level of some sort -- a predefined level like Minetown or the Castle, or a Gehennom maze, or something like that. We'll skip those -- they've all got their own quirks.

For a normal dungeon level, Nethack will create a bunch of rooms by examining the remaining free space in the dungeon level, and then trying to choose a random room-sized rectangle. It will repeat this process over and over until it can't find a suitable location to place any more rooms. During this process, there is a chance that one of the new rooms being created will be a vault. All of these rooms are placed in a list.

Then, the list of rooms is sorted by the x-coordinate of the left side of the room. To envision that, look at a map of a Nethack dungeon, and label the room furthest to the left 1, the next furthest 2, and so on. This is important for determining which room gets attached to which. (You may actually want to try drawing this out on paper so you can follow along.)

Then, assuming that both up and down stairs are to be generated for this level, a random room is selected for the downstairs and the down staircase is placed within that room. A random room is selected again for the upstairs -- if it happens to be the same as the room selected for the downstairs, then the next room in the list is selected instead, and the up staircase is placed within that room.

continued...

Then, we get into the code that connects the rooms. The magic there is done by the join() function, which tries to connect two rooms with a corridor, and deserves some explanation of its own. First it determines which walls of the two rooms the doorways should be in, and finds locations for the doors. Then, it finds a path for a corridor between the two doorways. When that is done, it places the doorways -- possibly making them secret doors, or trapped, or locked.

Finally, it changes a per-room value (the value in the smeq[] array, for those reading the code) so that the two rooms have an identical value. To picture this, if you labeled the rooms starting at 1, and then connected rooms 1 and 3, you would change the label in room 3 to be 1 as well. If you then connected rooms 3 and 4, then the label in room 4 would get changed to 1 as well. Eventually, if you kept connecting more rooms that way, every room's label would be changed to 1, at which point you would know that every room is connected to room 1, either directly or indirectly.

(The effect is rather easier to see than to explain, so if that did not make sense, try drawing it out.)

So back to connecting the rooms. The first step is to try to connect room 1 to room 2, room 2 to room 3, room 3 to room 4, and so on. If that succeeds, then you have a dungeon that is fully connected (albeit a bit dull) -- you can start in any room and get to any other room. The game may fail to create some of the corridors (due to not being able to find a good place to put them), or may randomly stop creating corridors, so at this point the dungeon may not be fully connected, though.

Next, Nethack will check if the label on room 1 and room 3 is identical, and if not, then try to connect those two rooms. The same with room 2 and room 4, room 3 and room 5, and so on.

Next, Nethack starts at room 1, and then checks the label on every other room. If it finds a room that has a different label than room 1, then it connects those two rooms. This repeats until every combination of rooms has been checked, or until all of the rooms have an identical label. At this point, all rooms on the map ought to be connected to one another.

Finally, a random number between 4 and (4 + the number of rooms) is picked, and Nethack picks that many pairs of rooms and connects them, which simply gives us some extra corridors between rooms.

At this point, the general layout is pretty much complete. Nethack then places niches (the one-square closets outside of some rooms), fills the vault with money if one was created, turns rooms into special rooms (shops, temples, and so forth), places the extra staircase or portal for branches, and then places monsters, traps, gold, fountains, sinks, altars, graves, statues, boxes and chests, graffiti in the dust, randomly placed objects, and gold and gems which are buried in the floor or walls.

So, that's pretty much it, at a high level. It's not exactly pretty, but it does get fairly good results.

What sort of circumstances would these be? In an RPG, they would be tied, naturally, to the player's experience level, thus ensuring that a level one party never finds itself face-to-face with a level twenty monster. In those action games, such as the exploration platformer, that draw on the most attractive feature of an RPG-- the player gradually becomes stronger and stronger as the game progresses-- the determination of which tier(s) to draw on would be made based on the player's accumulated power.

***Monster's Den: Book of Dread* only pits the player against randomly-generated enemies at the player's experience level.**

In games in which the nature of power is more tenuous-- that is, games in which the player gets a spike in power by gaining a power-up and might

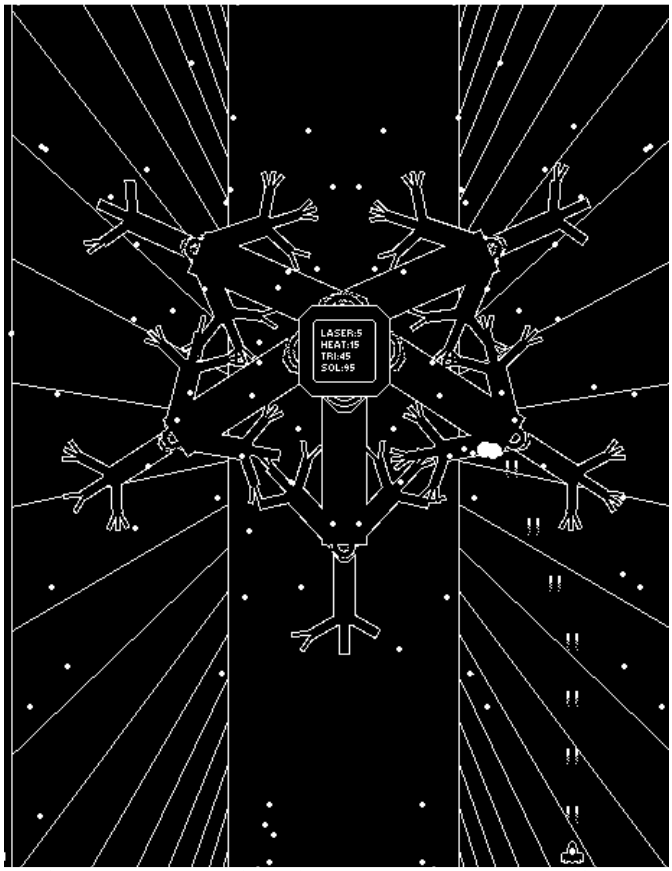
lose that power or swap it for another one-- tiers might be tied to those various power-ups. There is a danger in that, however; I can imagine a canny player going out of their way to avoid power-ups so that they can also avoid harder enemies. One way around that situation would be to tie the tiers to the player's progress in the game; as they complete more levels, tougher and tougher enemy types await them. There would, of course, need to be *enough* different enemy types to go around. Otherwise, it would defeat the purpose of creating enemies randomly in the first place.

It would also defeat that purpose if enemies consistently spawned at the same location. Whether a given tier has two enemies in it or twelve, if the player knows there's always going to be an enemy just after the second pit, he's going to play with that information in mind; he's not learning a skill but is merely learning how to avoid or deal with that specific instance. Just as you'd need multiple (and viable!) enemy types, you'd need multiple and viable spawn points.

You could take your level and litter it with spawn points that each have a fifty-fifty or even a one-in-four chance of spawning an enemy or not spawning an enemy; that would keep the player on his toes. But what if, defying the odds, *every* spawn point tested positive? The player would be up against an army of enemies. That's bad for the player, not to mention the slowdown it'll cause for your game. Not to mention the possibility of every spawn point coming up negative. Luckily, there are at least three ways to fix this.

Though they're not randomly generated, the presence of these "reapettes" in *Kid Icarus* often result in slowdown: a cardinal sin in game design.

One is to set a limit on the number of enemies that can be generated in a given area: four or six or eight, depending on your particular game. While this works extremely well for some genres (the shmup comes to mind) for others it can get fairly predictable; the player will catch on quickly and begin counting enemies.



Choosing a random boss in Cactus's revelatory *Fractal Fighter* often results in a level of difficulty few players are capable of meeting. While that's part of the game's charm, for most games this would be far too overwhelming.

Which brings us to our second solution, which also brings us back to tiers: assign each enemy a point value. Say, this goblin is 1 and that warthog is 2, et cetera. When the game generates its enemies from a given tier, it sets a maximum accumulated point value as well. For example, if only ten points worth of enemies can be in a given area, that could mean ten goblins or five warthogs or a mixture of both-- or even less if you add more powerful enemies to the mix. And then if this point value is also determined psuedo-randomly (say, a range between ten and twenty points) it also shakes things up considerably.

There is still, however, the question of pacing: how do we ensure that all the enemies aren't at the beginning of the level or its end, or that at any rate they're not spread out too intermittently? This brings us to our third solution: divide each stage into smaller themed chunks, or "mini-stages". Each mini-stage would have its own spawn points, its own determinators as to how many enemies and of what type can possibly occupy its environs.[3]

The mini-stage approach would also make it easier if you wanted to randomly generate the terrain as well; rather than having the program attempt to create an interweaving path across the length of dozens of screens, you would just ensure that a couple of clear paths exist over the course of one, two, or three screens. It would also allow you to exhibit more formal control over a given stage, thus minimizing the sometimes startlingly ugly effect randomness can have on a game.

Or, let me put it another way: action games aren't completely about enemies. They're also about obstacles, about tricky jumps, about falling blocks, about puzzle rooms, about bonus areas, about rest areas, about so many other things. Rather than putting information about all this into your program and hoping that after it weighs a few dozen variables and rolls an imaginary die a few hundred times that it'll all come out hunky-dory—that, for example, the game won't give the player an impossibly long chain of falling blocks with enemies spawning all over the place-- you could assign different challenges to different types of mini-stages: this one is enemy-heavy, this one is all about getting past those falling blocks, this one is a rest area or a checkpoint, et cetera. You could then arrange them in a sequence: rest, enemy, rest, enemy, falling blocks, checkpoint, really tough enemies, puzzle room, goal.

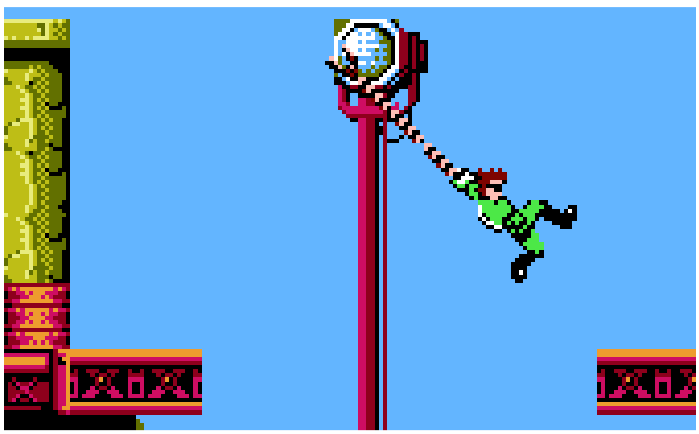
Or, hell, let's take this all the way: once you've determined that the first, last, and checkpoint areas are always "rest" areas, you can have the computer arrange the other elements randomly as well. In such a way, while the player would know what sort of challenges to expect in a given level, they won't know quite when they'll face them or quite what they'll be like. They'd be developing real gameplay skills instead of a situational simulacrum.

[3] For Game Maker users, this doesn't *necessarily* mean dividing the stage into multiple rooms (though I find that helps and actually recommend that approach). One could simply create an invisible object between two mini-stages; when the player object collides with that object, the next section will spawn its enemies, make its determinations, et cetera.

A Random Platformer?

But, as we all know, it's not as simple as all that. If it was, dozens of good randomly-generated platformers would have been made by now. And while randomly-generated power-ups and randomly-generated enemies in random spawn points are not precisely revolutionary concepts, the fact remains that no one's really ever tackled the Holy Grail, a rouge-like of Mario games: a platformer with randomly-generated level design.

Broken up into mini-stages or not, the genre still provides a number of formidable obstacles to any attempts at randomizing it. The first example that leaps to mind is, well, leaping. Precise jumps from one platform to the next, often over a bottomless pit or bed of spikes, is one of the genre's hallmarks. Even a game like **Bionic Commando**, a platformer that deprives the player of the ability to jump, is still about traversing over pits and moving effortlessly from platform to platform.



Let's say that our program sets a limit for the length of a bottomless pit or expanse of lava, one that the player can traverse. If that was the case, every jump would be merely perfunctory; push the jump button and a directional button long enough and the player will get over it. There's nothing in that that tests the player's timing and precision-- the whole point of bottomless pits in the first place.

The trickiest jumps are those that require the player to jump to a platform hanging over a pit. Okay: so how do we make sure that the game doesn't put those floating platforms too high? How do we make sure they don't put them too high so that we can't jump on them but too low so that we can't jump underneath them, either?

How can we ensure that the computer will create a series of floating platforms that must be navigated to reach the end of a stage or even a bonus area? How do we ensure that a bonus area isn't too hard to get to nor too easy?

Or, to really zero in on the question behind all these questions, let's look at the jumping block puzzles in the **Mega Man** games, in which the player must navigate a series of disappearing-reappearing blocks.

These sections are tight (and sometimes frustrating) platforming at its best. The blocks are spaced apart *just-so*; their appearances are timed *just-so*. Indeed, many of the best sequences in platforming games are those that depend on everything being arranged and calibrated and *just-so*.

The question, o best-beloved, is can a computer, a program-- no matter how many parameters it's been given, no matter how skillfully it's been finessed-- create something that's just-so? Can it ever make the same judgments as a human being? The more parameters we add to it, the more factors it has to take into account, the less random the end result is going to be, and if that's the case, what's the point of calling it a "random" platformer in the first place?

I am, as the reader might have noticed, somewhat skeptical about the chances of a truly random platformer; in many ways, the things we associate with the genre work against it. And yet it is the genre, especially in its 2-D form, that most egregiously depends on learning routes and preparing for specific challenges whose solutions can be learned through trial and error rather than actual challenges requiring thought and skill. If any genre could make use of a little randomness, the platformer is it.

But maybe a little is enough: maybe random power-up drops and enemy spawns are enough to keep us on our toes. In those platformers that depend on collecting a certain item, maybe the item location can change from one play-through to the next. Maybe these little things will be enough.

But maybe there's someone out there, a young game designer, who knows how to make it work, who will give us that holy grail, who will make a platformer that marries the artistry and just-so-ness of the genre with the challenge provided by the rogue-like dungeon crawler. Such a game might revolutionize the genre and the art form. Such a designer will require ambition, intelligence -- and perhaps even genius.

I'd like to think that such a designer is out there; I'd like to think that somewhere down the road, I'll get to play such a game.

Sprite Substance
by Tom Russell

It's not often that we discuss graphics here at the Quarterly; I'm still firmly of the opinion that how it looks doesn't matter nearly as much as how it plays. Sure, you can see all the pores and marvel at all the stubble lining the grizzly face of the gun-toting space marine in This Summer's Big Shooter, but in the end it's still a shooter and I'm still not having a whole lot of fun.

But that's not to say that the visuals are wholly unimportant. Certainly, one should avoid flat-out ugly visuals and, just as every game play element must fit with and complement all other game play elements, the various visual elements should look like they fit together. Place that carefully rendered space marine in a bright bouncy cartoony wonderland and it's going to seriously stretch one's credibility; likewise, *il fratelli Mario* would look out of place throwing fireballs at one of the ginormous aliens waiting at the end of each level of **R-Type.**

But this is still discussing graphics as the icing-- the scrumptious-looking exterior that hides what is really, at its essence, a set of rectangles and geometric shapes. The visual element has a much more important function-- and that is as a game play element.

Understanding Space

It is the visual element of a video game that relays to the player ninety-five percent of the information required to play the game. It is the visual element that tells us *what's* on the screen, *where* it is on the screen *in relation* to other things on the screen, and helps us to estimate *when* it will be somewhere else. It is a game's graphics that helps us to understand space and time.

Space and time-- and spatial and temporal relationships between objects and events-- are, as we've discussed before[4], the primary non-mimetic sources of meaning in the video game art form. It's vital that we, as game designers, never underestimate just how insanely important and crucial these relationships-- and the clear statement of these relationships using visual (and, to a degree, acoustical[5]) elements-- are to the moment-to-moment action of playing a game.

For example: let's say that (surprise, surprise!) a platformer confronts the player with a bottomless pit. How does the player know where the pit is? The spatial relationship between the player's character and the pit is spelt out in visual terms. How does the player know if he can clear this pit? The player, having made previous jumps, has some vague awareness of the distance and height his character can traverse while jumping; this he witnessed in visual terms. Perhaps he must do a running jump. How does he know when to press the jump button? When he's at the edge of the pit? But he's running; he might not be able to verify and process that he's reached the edge in that split-second between reaching the edge and falling. He'll have to estimate the approximate time that he'll reach the edge and be ready to press the button at that time; that estimate comes from an intuitive knowledge of his character's walking and

[4] cf. our fourth issue's cover feature, "Boss Battles", specifically pg. 30-40.
[5] See sidebar.

running speeds compared to the distance between himself and the pit. That is, it comes from the clear and visual statement of temporal relationships.

What About Sound?

I wrote an article for *Game Maker Technology Magazine* called *Visionary Sound*, in which I pointed out that most games only use sound as an "echo"-- as only another indicator of something we're already been told through visual means. But it doesn't have to be that way.

Sound design can "tell us something that is not immediately present in the visuals. One example is the use of music with an accelerated tempo to indicate that the player is running out of time in platformers such as **Super Mario Bros**. While the timer is visible on-screen, it's not the kind of thing one is paying attention to, especially if the sound is muted. With the sound on, though, it alerts us with stunning immediacy that we had better hurry it up.

A great example of visionary sound design is to be found in **Shush**, a surprisingly addictive game by Barry Atkins. Basically, the game tasks you with feeding coloured worms to correspondingly coloured and very noisy birds until they fall asleep. The size of each individual bird changes accordingly with each correct or incorrect worm, alerting you *visually* of its *individual* progress. As the birds fall asleep, they become less noisy, and so you are *aurally* kept abreast of your *over-all* progress. Combined, the two aspects (the visual and the auditory) give you a clear picture of where you are in the game, but each aspect *pulls its own weight*. Either one on its own can't get the job done."

All this sounds marvelously complicated, and in a strange way, it is; it's a matter of an eye perceiving several different spatial and temporal relationships at once, a brain processing them, and a finger or thumb pressing the button. And this happens over and over again, dozens of times per second, the brain, eye, and hand constantly reacting to new information. That it seems simple and intuitive has less to do with the wonders of the brain-- not that I would attempt to sell that magnificent and fascinating organ short-- and more to do with the visual nature of the art form.

Imagine, for a moment, a parallel universe in which this art form about which we are passionate took a completely different form. Instead of video games, let's imagine an interactive art form called touch games. (Now, now, stop the sniggering. Yes, you. I'll be seeing you after class, and then you won't find it quite so funny, will you?)

With a touch game, the player is given different tactile stimuli: different kinds of vibration, for example, each indicating a different kind of information about the state of the player and his game and prompting a different response. A danger vibration requires the press of this button; the power-up vibration is cause for jubilation. So far, while it might seem a little strange, our parallel universe art form isn't so bizarre as to be beyond our comprehension.

But it's not that simple. Because just as our video game art form doesn't hit us with one stimuli requiring one response at a time[6], their touch game art form gives the player nearly a dozen different bits of information concurrently. Enemies, power-ups, pits, HUD (TUD?), environmental hazards-- all of them represented by a different type of vibration or tactile sensation. Let's assume, for the sake of argument, that the player of these games would know what each vibration signified, that they would be familiar with the differences between each enemy type and its frequency.

That still leaves the question of how a player could *process* all that information simultaneously. And that's where our parallel universe and its arm form collapse on themselves. The only one of our senses that can take in and sort so many different kinds of information and relationships simultaneously is that of vision. It is, ironically, the only sense that allows us to experience a game in an immersive fashion. Tactile controller vibrations can give us goosepimples and a great sound mix can create a

[6] Well, unless it's a music game.

palpable feeling of atmosphere, but only our eyes can help us understand and interpret the virtual world we are in and the possibilities available to us.

Eyes are important for gamers. Don't let Salvador Dali anywhere near them.

That's one reason why I'm such a proponent of games that take place on a 2-D playing field, particularly platformers and action games. The 2-D scrolling approach allows for a crystal-clear understanding of spatial relationships. Anyone who has ever missed a jump in a 3-D platformer but sworn that they should have made it or anyone who has ever panicked and whirled the camera around trying to find that enemy that keeps hitting you are probably sympathetic if not simpatico. I would like to clarify, however, that I have played and enjoyed a number of superb 3-D games in which these problems have been greatly minimized if not precisely eliminated, thanks to the technical skill and design artistry of their creators.

Whether your game is 2-D or 3-D, platformer or shooter, boxing sim or match-three puzzle, the clear statement of these spatial and temporal relationships through visual means are not only important but your duty. Do not, under any circumstances, shirk it.

Intuitive Art Design

Just as it is important that we clearly state the spatial relationships between different objects, it is important that we clearly differentiate between these different objects. The player should not only be able to tell one enemy from another, but they should be able to tell the difference, intuitively, between an enemy object and a helpful one. A player should never, for example, run to pick up a new power-up dropped by a recently deceased enemy only to discover, to their detriment, that it was, in fact, a bomb. (And yes, I have encountered this before in more than one freeware game. Otherwise, I wouldn't feel the need to mention it.)

Similarly, a player should never discover (to their horror or amusement) that they have just attacked an NPC because they mistook it for an enemy, and they certainly should never have to take damage from a monster they mistook for an NPC. (And, yes, I've encountered this too, though I've seen it both in freeware and commercial games.)

Such fissures are not annoying, but game-breaking. They conspire to create the exact opposite of an immersive experience. And, sadly, they are all too easy to avoid.

When creating the graphics for a game, the designer should make a number of stylistic choices: they should determine visual qualities or traits that have positive and negative associations and once they have settled on these traits, they should be applied consistently through out the game. In other words, the designer must create a *visual language* for the game that the player can quickly pick up on and trust.

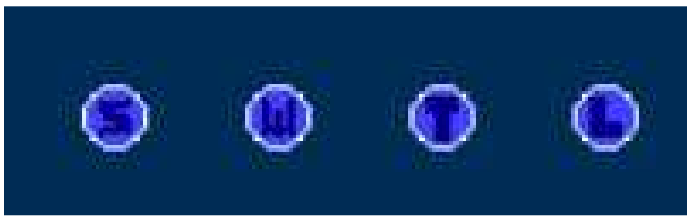
Let's look at a couple of examples. First, let's tackle that pesky problem of trying to talk to enemy characters or trying to attack NPCs. If our game features a number of different enemy types and a number of different NPCs existing in the same game space (instead of, for example, being segregated into "action" and "town" areas), the player needs some quick-and-dirty rules to help him sort out the difference. In some games, this is actually quite simple: the enemies might resemble animals while the NPCs look like human beings, or the enemies might look like robots while the NPCs resemble animals.

> You must create a consistent visual language that the player can quickly pick up on and trust.

What if, however, your game takes place in a more realistic milieu, comparatively speaking? Instead of fighting animals, demons, or robots you're fighting other human beings? How do you visually demarcate the difference between your NPCs and your enemy characters?

What we need is a trait-- some kind of visual rule that holds true for almost all objects within a certain group, something that screams "Hey, I'm an enemy!" For example, if all enemy characters carry a weapon or wear battle armour, while all NPCs are unarmed and dressed in peasant garb, it becomes considerably easier for the player to tell the difference between one and the other.

The same applies for our example of a bomb mistaken for a power-up. If all power-ups have a visual property in common-- for example, if they're all yellow rectangles or circles with some sort of insignia on them, or if they are all roughly the same size-- then the chances of a player running into an eminent explosion are greatly diminished. Power-ups especially lend themselves to a certain degree of iconography and geometric abstraction in visual design.



Let me restate this, because it is absolutely crucial: each class of object-- power-up, enemy, NPC, environmental hazard, whatever-- needs at least one distinct visual rule or motif assigned to it, and each object within that class must conform to that rule or motif. To deviate from your design is to do so at the risk of confusing and alienating the player.

Those of us attempting to design games in this day and age are fortunate in that a certain "universal" visual language has been developed already; it is a rich tradition that we ignore at our peril. We know, for example, that spikes and bottomless pits often represent instant death, that hearts and first aid kits regain hit points, that smaller statues of ourselves or busts of our head bless us with an extra life, that a flashing character has just taken damage and/or is temporarily invulnerable, that a flashing object is likely to explode, that lampposts and swirling swatches of stars save our progress, and that exceptionally large enemies are likely to be boss characters (for starters). There is no shame in drawing on these powerful symbols, symbols that are all the more powerful because they are quickly and intuitively understood by players-- the whole reason to create a visual language in the first place.

Playing Fair

I remember a certain very well-regarded game that was created using Game Maker that I simply couldn't stand. I will not mention it by name, as that serves no useful purpose, but I can confide that it was an unusual sort of puzzle platformer with a highly truncated time limit. The time limit wasn't the problem.

The problem was that the "puzzles", if you could call them that, went so far beyond being merely non-intuitive that there was no way for a player, regardless of their intelligence, to puzzle them out. Some blocks disappeared underneath you as you tried to walk on them, some blocks fell from above and squashed you, some walls contained hidden passages; spikes killed you every time you touched them except when there was a row of invisible blocks situated on top of them.

It wasn't that the game featured these hazards and secrets-- such things are actually fairly common for platformers especially. It was that there was no way to know a block would fall out from underneath you until it did so; there were no clues that some blocks were about to fall on top of you or that there might be a secret passage nearby. In short, it didn't play fair *with* the player but rather unfairly *against* the player.

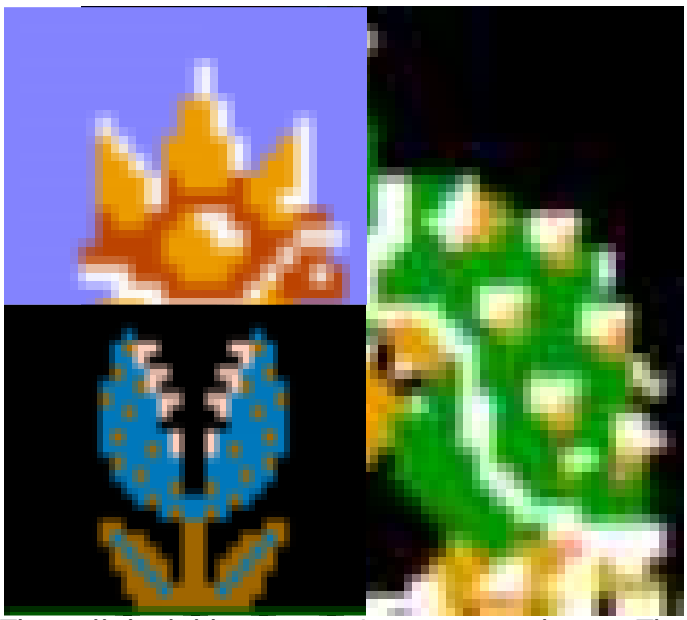
That game's difficulty wasn't the result of giving the player all the information he or she needs to conquer the challenges set before them; it was the result of the game's designer showing off how much smarter he was than the rest of us peons. That game has been hailed for its incredible game design; it is, in fact, a prime example of terrible game design.

Platforms that have some kind of special properties should indicate those properties *visually*. Falling platforms should look like they're going to fall; platforms that zip up to the ceiling when the player lands on them should look like they might go up (maybe an arrow?); "quicksand" platforms should look, well, like quicksand. Simply reusing the same sprites you use for normal platforms is not only not an option, it just isn't kosher. *A player should never be penalized because he lacks information.*

A game designer has more leeway, I suppose, with "secrets"-- with invisible platforms and hidden passageways. The whole point is that at first glance they don't draw attention to themselves. But there's the key phrase: *at first glance*. The player should, on their second or third glance, be able to figure out that something's amiss. An item or platform that appears out of reach might be all you need to tip the player off, but there needs to be *something* there. A puzzle-- especially a "mandatory" puzzle, one that must be solved if the player is expected to progress to the next stage or area-- that doesn't give the player all the information needed to solve it is no puzzle at all: it's like reaching the end of a mystery novel and discovering that the detective has fingered the culprit using a clue that was never mentioned in the text.

Not to belabor the obvious point, but playing fair is the name of the game, and one of your best tools to ensure fairness is the use of visual elements. Not only can you use visual clues to indicate which platforms are safe and which are faulty, but you can use them to indicate different properties possessed by certain enemies.

For example, there are three enemies in **Super Mario Bros.** that cause the player to sustain damage if he tries to stomp on them: the spiny, the piranha plant, and the boss enemy, Bowser Koopa. Now, what do these three objects have in common?



They all *look* like you can't stomp on them. The spikes lining the shells of Bowser and the spinies and the teeth of the piranha plant look distinctly unstompable. And this is the key to imparting information, and thus substance, through one's spriting: *things should look like what they are*.

> A player should never be penalized because he lacks information.

If, for example, stomping on a Bullet Bill caused the player to take damage, it wouldn't make any sense. There's nothing about that sprite that makes him look unstompable. He does, however, look fireproof-- a trait, both visual and actual, that he shares with those pesky buzzy beetles, indicated by their shiny black exterior. If the lowly goomba had been fireproof, it would have been confusing and appeared arbitrary, as he looks distinctly flammable. (And delicious. Mmm. Portobellos.)

Goomba, post-stomping. Photo by "pfctdayelise".

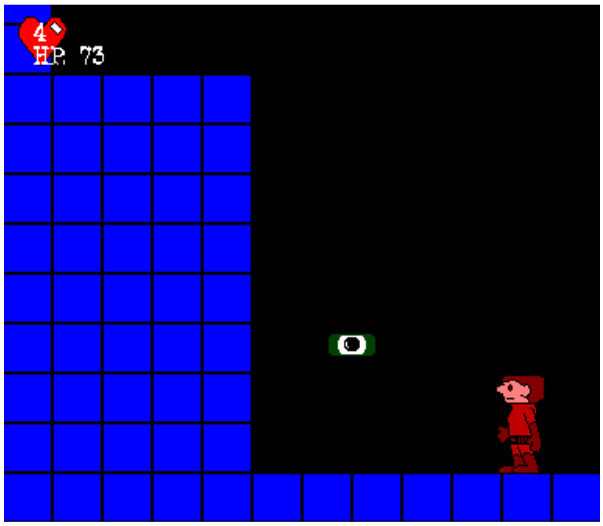
Consider some of the enemy design in the first game featuring our cover star's rodent rival, **Sonic the Hedgehog**. The caterpillar enemies that we encounter in the Marble Zone are lined with spikes; attempting to spin attack those spiked segments will cost our azure atelerix albiventris some of his rings. Compare this, however, to the bomb enemies in the Star Light and Scrap Yard Zones. Attempting to attack them will only hurt you, and yet there's really nothing about them that says, "Hey, you can't attack me!" The player can only find out the hard way.

A player should never be penalized because he lacks information. It's important in action games which give the player multiple avenues of attack that the visual design of a given enemy indicates which attacks it is not susceptible to. If an enemy is unkillable, or a kind of environmental hazard--

for example, a crushing machine-- the player should not have to attempt to attack it in order to find out.

Which reminds me of one of my own game design mistakes. (The reader who thinks I am pretentious or a "Know-It-All" should be advised that almost all of my ideas about game design come from doing it the wrong way many, many, many times before.) In my very first Game Maker game, **Orfee Firewalker**, which is mercifully no longer available on Teh Interwebs, I made a number of the mistakes I rally against in the paragraphs above. Non-intuitive secret passages? Check. Deadly traps that gave no indication as to their whereabouts? Check. Invisible platforms that no one found except the man who put them there in the first place? Check.

And then there was this guy, Ocully. He was a little floating eye that moved along this erratic, looping path.



But if you hit him with your weapon (the "soul ball"), he doesn't die like every other enemy. No, he splits into *two*! Or three! Or four!

Besides resulting in severe slowdowns when the player had a particularly itchy trigger finger, it just didn't make any sense. The eyeball didn't look like it was invulnerable to the player's attacks; in fact, it looked downright squishy. It also didn't look like it could clone itself, either, though that opens up the question of, how do you design an enemy that looks like it'll multiply if you attack it?

Well, besides using the legendary hydra-- cut off one head and, as Baron Von Strucker liked to say, two more will take its place-- one could modify the Russian Matryoshka, or nesting doll, so that each time you defeat one, *two* smaller versions are born. And though my memory fails me on the specifics, I do recall at least one game which presented the player with a ball of goo that spawned other balls of goo if attacked or provoked. And that felt right at the time-- its gelatinous properties seemed to mimic an act of asexual cell division.

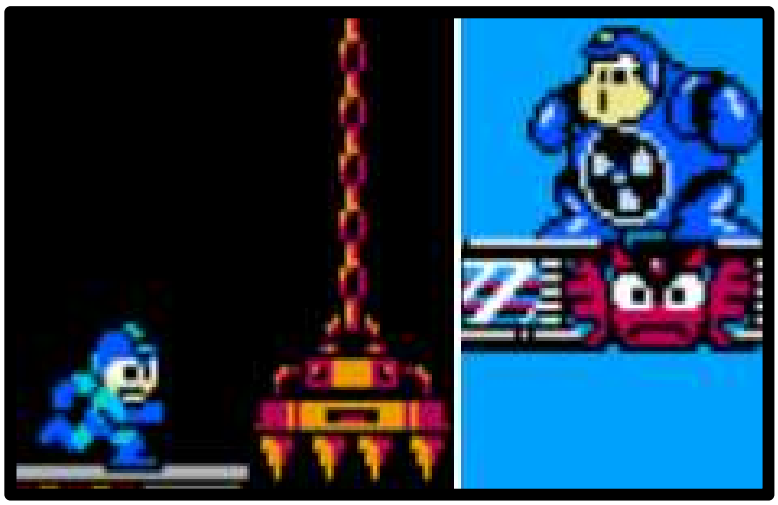
So there are always answers, if you're willing to think a little more deeply than I did with **Orfee Firewalker**. At the very least do not duplicate my mistake of creating an invulnerable enemy patterned after one of our most fragile and vulnerable body parts.

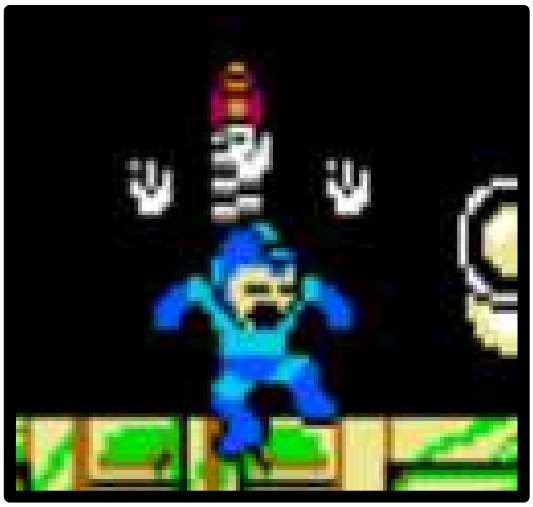
In fact, a cursory glance at good games will reveal that the vast majority of invulnerable enemies look, well, invulnerable. Tough shells, scaled armour, and shields are the norm; in games with enemies patterned on organic life, invulnerable enemies will often look robotic or like beings of pure energy. Indeed, one way to differentiate between an enemy and an environmental hazard is to ensure that the latter is more abstract in appearance. That is, while enemies-- human, animal, robot, demon, or whatever-- generally have identifiably anthropomorphic features (such as eyes and limbs), hazards lack these qualities. The crushing machines in many games (such as the Metal Man stage in **Mega Man II**) and the firebars in **Super Mario Bros.** emphasize rough geometric shapes. Coupled with nonorganic textures-- in the case of our two examples, the crushing machine is extra-rusty and drab while the firebar is a bright fiery red, orange, and white-- this use of abstraction clearly indicates that no matter of stomping or shooting will deter these inhuman death traps in any way, shape, or form.

But, of course, that's only if the game's enemies *are*, like most enemies, organic-looking; if your game also features highly abstract enemy design, going the same route with the environmental hazards is a no-no-- just as the player should never confuse a power-up for a bomb or a stable platform for a death trap. *Things should look like what they are*. Above all, they should make some kind of sense.

(CONTINUED ON PAGE FOLLOWING...)

The "Donut" platforms in the later **Mario** games, which fall once you step on them, are clearly differentiated from non-dangerous platforms by sporting their unique look.





The hazard on the left emphasizes abstract geometric shapes, while the enemy on the right is anthropomorphic. Note also that while enemies are decked out in bright primary colours, the invincible hazard uses a more subdued palette.

It's important to remember that visuals serve as an indicator of an object's status as well; here, the player taking damage is shown not only graphically-- the change in the player sprite-- but also *spatially*: he bounces backwards.

The visual language of **Shadow of the Colossus**: enemy parts that can be climbed (hair) and those that cannot (armour).

Pro Killer Man uses simple colour coding to differentiate between the player (in black), the target (in white), and the cops (in blue). Even amid a throng of "civilians" it's possible to make out where you are and where you need to be. It's a remarkable achievement in visual language.

A Word About Hierarchies

What we're really doing in creating a visual language and assigning visual indicators to certain object traits is we're classifying things in such a way that the player can grasp a complete knowledge of the game's system in the blink of an eye. And since we're classifying things, we should have a word about hierarchies.

Not every enemy or object is or can be incredibly and dramatically different from every other type of enemy or object. Sometimes they'll have enough in common with a certain object that the best way to appraise the player of those similarities and to highlight those slight differences is to resort to a palette swap. There's no shame in this technique, as long as you use it in the manner I just described and you don't overdo it.[7]

But, as always, you must be consistent. If the red type of an enemy is weaker than the blue type of an enemy, or if the red one has a weakness to an ice attack while the blue has a weakness to a fire attack-- then all red enemies and all blue enemies should share those properties. Red and blue become indicators, become shorthand for those properties, become a part of your visual lexicon.

Consider again the example of the buzzy beetles and the Bullet Bills from **Super Mario Bros.** Their shiny black shell is shorthand for "fireproof" in the same way that spikes are shorthand for "unstompable". And this use is consistent throughout that game; black is always fireproof just as the firebar, podoboo, and Bowser's flames are indestructible hazards because of their fiery colouring and shape. If any flammable enemies used black colouring or if any destructible enemies used the fiery look, it would be confusing.

And so, if you're going to set up hierarchies-- in which different objects or the attributes of different objects are ranked by colour-- then, as always, *you must be consistent*. Everything must fit together: that's the core of game design.

Another common way to "rank" objects is by size. As a general rule, bigger is better, stronger, and tougher. A giant version of an enemy should be tougher than a smaller one. At the very least, you should avoid the inverse.

Power-ups especially lend themselves to size-based hierarchies. For example, in the **Mega Man** games, larger power dots restore more health than smaller ones. There's also the case of fractional-based hierarchies, in which the actual size of the power-up icon is roughly the same but what is contained within it changes: in the **Zelda** games, there are full-heart containers and quarter-heart containers; in the various **Teenage Mutant Ninja Turtle** games, as in life, a full pizza is better than half a pizza. I don't think a fractional approach would be entirely too successful if applied to other object types, such as enemies; I can't imagine feeling particularly threatened by two-fifths of a goomba.

One more word about hierarchies: they should not be based solely on hit points. If the only difference between your red enemy and its blue counterpart is that the blue one has two extra hits, you're not really doing your job as a game designer. Certainly, your blue enemies can have a higher threshold for damage than their red cousins, but they should have something else to distinguish them, however subtle: a different type of movement, an attack that controls more screen space. The trick, of course, is to make sure there's enough in common with other enemies in its group that the association formed by their shared sprite makes sense. You can have a red goomba and a blue goomba, but if the blue goomba throws hammers and leaps from floating block to floating block, you'll be better off making it a Hammer Brother and be done with it.

In Conclusion...

Now, a lot of what we've just went over is a trifle bit on the obvious side. *Be consistent. Things should look like what they are. Play fair; give the player all the information they need.*

But just because it's obvious doesn't make it any less vital and important. A number of game designers, myself included, either never learned these lessons or forgot about them completely. Hence the preceding refresher.

One should never neglect the basics on the logic that they're "simple". Simple as they may be, they are also the very foundation of good game design.

[7] Yes, **Mortal Kombat**, I'm looking at you. I'll be seeing you after klass. (kracks ruler.)

Miyamoto's Framework
by Paul Eres

On a forum post I mentioned that what most people consider a "game" today comes out of a system that Miyamoto was the first to systematize, and I thought I should expand it in an entry to work out the idea.

The idea is that virtually all games today fall into that framework, even though it's almost an arbitrary framework, the way that virtually all manga falls under Tezuka's framework even though his was almost arbitrary too.

The framework consists of these elements:

1) The player controls a character directly, and can move that character around. They press a direction, it moves in that direction This can be in 1D (platformers like Mario), 2D (overhead like Zelda), or full 3D (like flight sims). Sometimes the player isn't alive, they can be a block (as in Tetris), but there's something that you can "move" on the screen.

2) The player has to get somewhere, find something, or kill some enemy to win, sometimes with a time limit, but often without one. This is achieved by getting the player-moved thing to the goal through a long series of obstacles. Once done, it's the end of the game.

3) The player can die or lose by doing the wrong thing, such as falling into a hole or walking into an enemy or losing a battle. If they do so, they have to try again.

4) There are checkpoints or the game is segmented into spatial sections, stages, chapters, dungeons, or missions; it's not continuous, there are marked points along the way to the end of the game. If the player dies after completing one of these, they can continue from the start of the last one and try that section of the game over again. Everything in the game exists in space, with coordinates (in either 2D or 3D).

5) There's a gradual increase in the player's capabilities or items or weapons, the player gathers or purchases these along the way to the end of the game, such that at the beginning of the game the player's character is much weaker than that character is at the end of the game.

There are variations of this framework too. Sometimes the player controls a group of characters rather than one (as in a RPG). Sometimes the player does not control the player directly, but more indirectly (RTS and turn-based strategy). Sometimes the player can't die (adventure games).

These five principles are surprisingly consistent, even when one is broken, the rest are kept. *Ten thousand* great games have been made under this framework. They've become so prevalent and expected that anything not using these principles (or at least two or three of them) is generally not considered to be a game.

But, they're all almost arbitrary. There's no reason to use this framework and not some other framework. You probably could make *ten thousand* great games under five totally different principles, provided they worked together as well as these five do. Yet only a few truly great games don't use any of the five principles of the Miyamoto framework (some of Will Wright's games for instance), even though there's no reason why you couldn't have more.

Shigeru Miyamoto; photo by Vincent Diamante.

Now I'm not saying we should have goalless games or games without any principles at all, just that the principles can be very different from the dominant framework. Other principles could just as easily work, I'll go through the five above and explore some alternatives:

Why do you have to control a character or a set of characters, why can't you control something else, like the level itself, or like the weather, or something indirect like that, rather than something which essentially follows your commands like a robot?

Why do you have to "get somewhere" or "find something" or "kill something" to win? One once-popular alternative was to get a "high score" instead. But there are other alternatives too: why not "arrange a system correctly" to win, as in Will Wright's games? Or why not "bring peace to things that fight each other" or "get things which are peaceful to fight each other" or any of a number of other goals?

Why does the player have to go back and try again if they succumb to an obstacle? Why doesn't losing turn you into an enemy, what you were just fighting? Why doesn't it make the screen gradually darker? There are a variety of ways to lose and a variety of obstacles, and "going back to the start to try again" is only one of them.

Why do games need to be sectioned up in distinct areas and dungeons, why not a continuous area? Why have spatial maps at all? You can create a great game without a map, you can set it in "dramatic space" rather than "spatial space", or any of a number of other ways to organize the game's content, such as a file system, or the way the internet is organized (there's no spatial map of the internet, just one-way hyperlink warps). Why do the objects in a game need to have a physical location, why can't they be more diffuse, like spirits which are everywhere but nowhere?

Why a gradual increase in the player's capabilities, why not a gradual decrease? Wouldn't that be more challenging (albeit somewhat depressing in theme)? Or why not make abilities and actions contextual rather than having a stock set of them that can increase? Or why not give the player all of them at the beginning, but not tell the player how to use them, so that they have to gradually discover their own latent powers, by experimentation?

It's clear that the framework Miyamoto set up works, and that it was an enormous achievement to find a set of principles like that which worked so systematically and wonderfully. And, as I said, ten thousand great games have since followed that achievement. But there are very likely other configurations of other principles which could work out just as well, and an additional ten thousand great games could be made under a newly discovered system which works as well as this one does. We just haven't found those configurations yet! But once they are systematized they'd be just as powerful as his.

I think the reason most "experimental" games fail is because they simply try to break the five principles of Miyamoto without forming a new system of new principles. They're basically like the "revolutionary" anarchist who is against everything but doesn't have anything to replace it with, they tear down but they don't build up anything new, and even when they do use a new principle, they don't use *many new principles systematically*, which is what would really be required.

The number of possible principles and especially the number of combinations of these is almost infinite, but most combinations will fail, and only a few like Miyamoto's combination will work. By work I mean be flexible enough to support a large enough gameplay variety to allow thousands of distinct games to be made using the new systemization.

If you take Miyamoto's framework and just reject principles at random and replace them with new principles at random, chances are you won't wind up with a new system that works, you'll wind up with a mess. You can't just "reject goals" and "reject the player dying" and make something without them and hope it'll be great, you need to form a new systematic set of principles which work as cohesively together as his do.

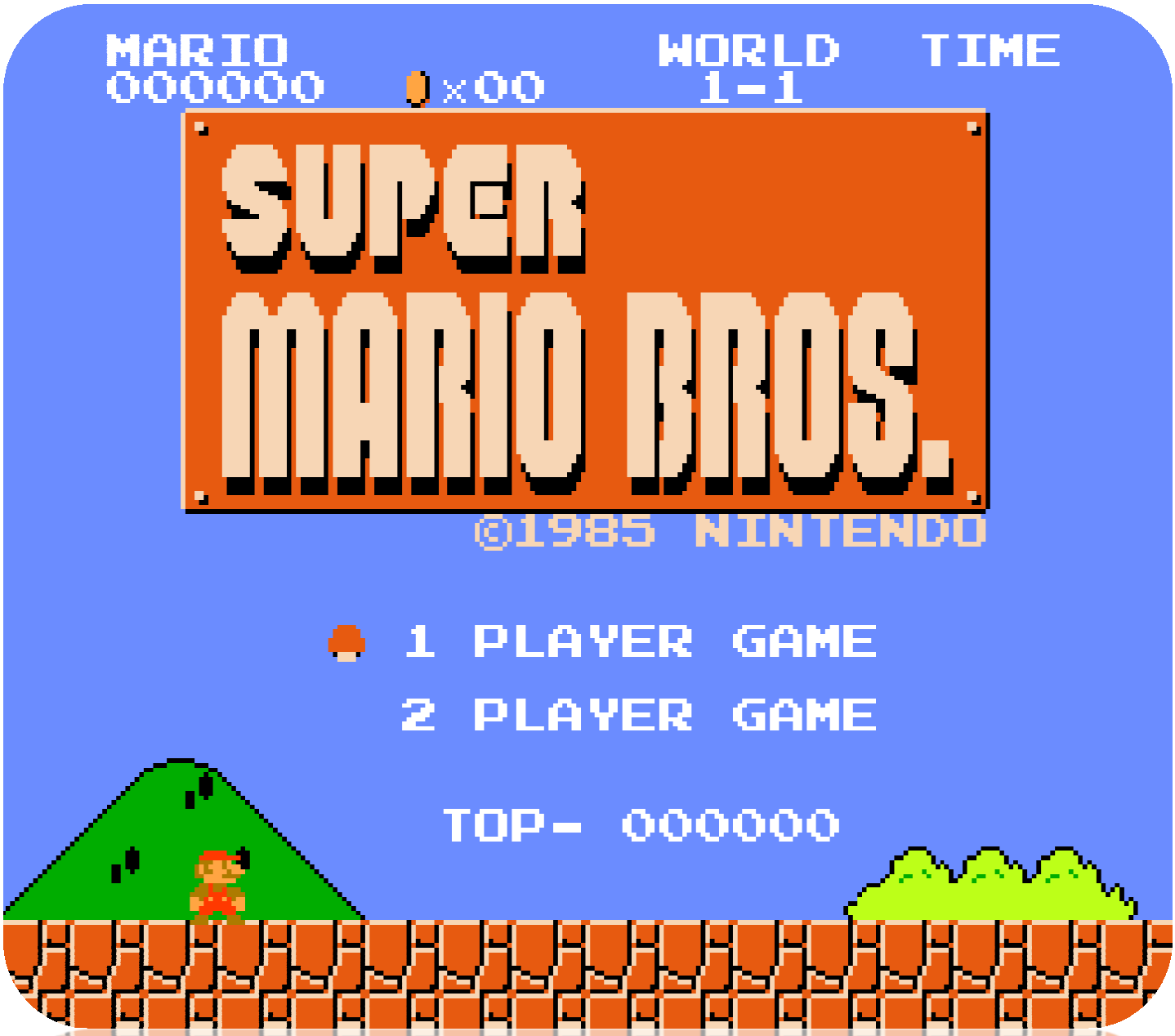
For instance, take the early efforts toward interactive storytelling, such as Facade or the Storytron. It's the beginning of a new system, it has one or two new principles, but not yet there, it's probably missing one or two other principles which it needs to truly compete with Miyamoto's framework. I can see it beginning to form, but it may take another ten years or so before it's systematized correctly. But I think that when it's systematized, its scope will be larger than Miyamoto's framework's scope.

Will Wright's "sim-system" framework has a small but interesting scope, because it allows potentially dozens or hundreds of distinctly great sim-system games to be made. But such games aren't for everyone, they're hard to pick up unless you are interested in the system being modeled, and they're inherently educationally oriented rather than dramatically or emotionally orientated, they're about the complexity of physical systems, which is a great thing to know about, but there are only so many complex physical systems to understand before you know them all, so I'd say the sim-system framework's scope is slightly smaller than the Miyamoto framework's scope, even though it might not seem so because it's less explored than his scope has been.

Will Wright; photo by Nathan Borror.

There are also more ancient game systemizations than Miyamoto's: puzzles, board games like chess and checkers and mah jong, card games like poker, and the like are one systemization first set up thousands of years ago involving tokens and objects with different values, and which was dominated over what we believed "games" to be for an incredibly long time. Sports, physical sports like baseball and soccer, are yet another systemization that existed in parallel to that one. Neither translates to computer gaming very well, although puzzles do so much better than sports, because they don't rely on physics, they rely on simple mathematics and geometry.

But those are just a few potential alternative game systemizations which have been discovered so far.



Super Mario Bros. needs no defenders. Everybody already knows it's a terrific game, a masterwork both of genre and form. There are thousands if not millions of enthusiastic appreciations in print and online about the game that many still consider to be the textbook example of how to make a sidescrolling platformer. So why the need for another? What makes this one so different?

If, indeed, it is a textbook example-- and I believe it is-- the question is begged: what can we learn from it? What does it have to teach us about level and enemy design, idea and execution, pacing and pleasure? What wisdom might be revealed by a closer and deeper study of this seminal game?

What follows attempts to be such a study. It is not a compendium of glitches, tricks, and tips or a walkthrough. You can find that anywhere. It attempts instead to be a deeper probe into the essential game design principles that flow beneath the surface, and at the actual nitty-gritty of how the game delivers on those principles.

Lesson # 1: Extrapolating from a High Concept

And just as the Japanese amuse themselves by filling a porcelain bowl with water and steeping in it little crumbs of paper which until then are without character or form, but, the moment they become wet, stretch themselves and bend, take on colour and distinctive shape, become flowers or houses or people, permanent and recognizable, so in that moment all the flowers in our garden and in M. Swann's park, and the water-lilies on the Vivonne and the good folk of the village and their little dwellings and the parish church and the whole of Combray and of its surroundings, taking their proper shapes and growing solid, sprang into being, town and gardens alike, from my cup of tea.

--Marcel Proust, *In Search of Lost Time*.

"[**Super Mario Bros.**] started with a simple idea," designer Shigeru Miyamoto once told an interviewer for *Business Week*[8]. He wondered "what it would be like to have a character that bounces around. And the background should be a clear, blue sky."

Everything else in the game flows out of that simple idea, just as a lifetime of memories, digressions, and observations spring out of a few cookie crumbs in Proust's cup of tea. Bouncing around dictates how Mario will attack his adversaries: he stomps on them from above. He smashes bricks with his head from below, gaining power-ups and disrupting enemies that way. He nabs coins in mid-air. He leaps over pits. The major challenges and basic actions of the game all revolve around Mario's "bounciness".

The disappearing platforms. The elevator lifts. The scales. The staircases and the flagpoles. The springboards. All revolve around jumping. Huge horizontally-scrolling levels, while not completely unique to Miyamoto, were far from the norm; they allowed more room to jump and bounce about than the single vertically-oriented screen of the arcades.

Because the game revolves around bounciness, it often features things to bounce on or to, generally offering a higher or lower path. This "branching out" is magnified by the presence of vines (going up) and pipes (going down) that lead the player to secret and bonus areas. This in turn is magnified by the presence of secrets: secret items hidden in blocks, invisible blocks, warp zones, maze-like castles, even the creation of fireworks.

Talking about **The Legend of Zelda** at the 2007 Game Developers Conference,[9] Miyamoto recounted the confusion that that game had engendered among others at Nintendo. Rather than make the game easier and more linear, he made it more difficult by initially depriving the player of his sword. He theorized-- years before our internet age-- that players would have to talk to one another, share secrets, communicate, and figure stuff out together. This would also, coincidentally, increase the "life" of a game and its hold over a player's imagination.

Though he was talking specifically about **Zelda**, the same theory is put into action in **Super Mario Bros.**, which was produced concurrently with its less linear cousin. The game is full not only of secrets but of content that have kept players chattering with one another to this day. Whereas earlier games, especially those produced in an arcade, showed players the entire game in the space of the first few (if not the very first!) boards and then repeated that content, **Super Mario Bros.** staggers the introduction of its enemies and obstacles. And it's not just a matter of palette-swapping; each enemy has unique movement and

8 http://www.businessweek.com/magazine/content/05_45/b3958127.htm

9 As reported by Kotaku; http://kotaku.com/gaming/gdc07/gdc07-liveblogging-nintendo-242670.php

attack behaviours, each one susceptible to some player attacks and not to others.

All this content-- an entire world of possibilities!-- springs forth logically and passionately from Miyamoto's deceptively simple idea of "a character that bounces around." This process of extrapolating from an intuitive high concept, of building from the center out, is not only the idea at the heart of the best video games, but is, indeed, the idea at the heart of the art form itself, one that's foregrounded in the very act of playing. To better illustrate this, and since we're talking **Mario**, let's take a look at the controller most of us first used to play the game.

Excluding start and select, there are two action buttons and a directional pad at the player's disposal. Yet in **Super Mario Bros.** the player is capable of, by my count, twelve distinct actions depending upon the context: he can walk, run, jump, perform a running jump, swim, stomp from above, hit a block from below, fire, kick, climb, duck, or go down a pipe. And that's, of course, folding one of the player's most context-sensitive offensive actions-- the use of the Starman-- into walking and running, while discounting showboat maneuvers like double jumps, wall jumps, sliding, and moon-walking. Twelve different actions from basically five different buttons. That is, of course, counting the left and right portions of the D-Pad as a single button; otherwise we have six buttons and sixteen actions, doubling those actions requiring movement to denote both possible directions.

But if you ask the average player for a quick knee-jerk answer to the question, "What do the buttons do in Super Mario Bros.?", the answer will probably resemble the following: "This one moves the guy, this one jumps, this one shoots a fireball and lets you run if you hold it." Each button, then, has a distinct (and intuitive) purpose or action and yet is capable, given the right context, of interacting with the game's system in a number of unique (yet still intuitive) ways.

That, in essence, is what game design is all about. Each rule or component of a system should be readily understood by the player and fit intuitively with all other rules and components. The essence of a good game can and should be grasped in an instant, just like the control scheme in Super Mario Bros. And yet the intersections of these rules, components, and features, like that control scheme, allow for a growing number of interactions, complications, and results.

Lesson # 2: Power-Ups

Mario ended up being too big, so we shrank him. Then we thought, "What if he can grow and shrink? How would he do that? It would have to be a magic mushroom!"

-- Shigeru Miyamoto, *Business Week*

No discussion of ***Super Mario Bros.*** *can be complete without talking about its extraordinary and elegant use of power-ups. And, indeed, I've discussed it quite fruitfully before in the very first issue of the* Quarterly. *Habitual laziness compels me to simply reuse that material here rather than try to come up with a new angle or appreciation of it. If you've read this material recently, please feel free to skip or skim over it. For the rest of you, the discussion begins by overlapping slightly with what we just discussed: extrapolation.*

The entire game is a marvel of imagination, extremely robust yet founded on an elegant simplicity. It starts with a few very basic rules. For example: if an enemy touches you, you lose a life, but if you stomp on an enemy, you kill it, scoring points; stomp on more than one enemy without touching the ground, and the points double with each successive enemy. Before even touching on the three power-ups, let's look at how much the game extrapolates from even this simple rule.

The first thing it does is create exceptions to the rule, enemies that can't be destroyed by stomping; stomping on a Spiny or one of the bosses would cause the player to lose a life! To destroy them, the player must utilize *another* enemy that can't be destroyed by stomping-- the turtle-like Koopa Troopa. If a player stomps on this enemy, he hides in his shell; the player can then kick the shell, sending it spinning into-- and destroying-- even "invulnerable" enemies.

And, ingeniously, the kicked shell follows the same rule as successive stomping: each enemy it destroys is worth twice as many points as the enemy before it. By scoring enough points in succession, whether by shell or by stomp, the player begins to win extra lives.

This twist, in turn, provides the setting for the most notorious and poorly-kept secret in the annals of video game history: the staircase in level 3-1. No matter how many times the player stomps on one of the Koopa Troopas descending that staircase, they will not be destroyed. And if you continually stomp on the shell without touching the ground, each stomp will double the points until you have one extra life, two extra lives, twenty, thirty, up to 128.

And this masterpiece of broken game design, which virtually ensures that the player can beat the game, is the end result of the rule with which we began, and the exceptions created for that rule.

The power-ups in **Super Mario Bros.** function in much the same way, created either as exceptions to a rule or as a tool to counteract those exceptions. For example: if an enemy touches you, you die, *but* if you are empowered by a Starman, which renders Mario temporarily invulnerable, the enemy is destroyed instead; if you have been powered-up to Super Mario or Fiery Mario, you will not die upon impact but be reduced to the non-powered regular Mario state.

The Fire Flower, like the Koopa Troopa shell, allows players to counteract the exceptions to the stomping rule; that is, the enemies that cannot be killed by stomping can be killed by a fireball or a well-kicked shell. A notable exception on top of the exceptions are the beetle and bullet-bill, whom cannot be killed by the fireball but are susceptible to the shell attack or the Starman. And some hazards, such as the bottomless pits and fire-enemies in the castle stages cannot be destroyed at all, making them the great equalizers.

This system of rules and their exceptions, and exceptions within the exceptions, helps balance the game and prevents the fireball from becoming infallible. If Fiery Mario could destroy everything with this attack, once the first Fire Flower has been claimed in 1-1, there would be no excuse to lose a single life and the game would have no challenge.

At the same time, if the situation was reversed-- if the Fire Flower was a very specialized attack, not working on the vast majority of enemies but only on a few exceptions-- it would be an utterly worthless power-up. Not only would its actual value be greatly diminished, but its perceived value, the feeling of an increase in power on the part of the player, and the desire to achieve that power, would be diminished as well.

A good power-up dramatically changes the game dynamic. The game-play before gaining the power-up and the game-play after is markedly different, and it should be different for the better. Hence the name, power-*up*. At the same time, the contrast cannot be so strong as to make the "before" game boring and unrewarding; neither can a designer depend solely on the power-up to provide excitement. It is frustrating for a player to feel weak, powerless, and vulnerable in the "before" game. In **Super Mario Bros.**, the player is able to run, jump, dodge, stomp, and kick turtle shells before coming across the Fire Flower. All fun stuff, all very athletic, and the player still feels strong and powerful. But with the addition of the Fire Flower, the player can toss fireballs-- he can strike enemies from afar and even from above without risking an accidental misstep that results in his own destruction-- the player is and feels *more* powerful.

And, in fact, the perceived value of the Fire Flower power-up is boosted by the presence of the Super Mushroom, which transforms the player into the double-sized Super Mario-- a transformation that *must* take place before the transformation into Fiery Mario. Furthermore, being touched by an

enemy in either of these two powered-up states will reduce the player to the regular (or "small") Mario state, necessitating finding another Super Mushroom and another Fire Flower. Like our game of four-square, the perceived value of the Fire Flower is increased because of the hard work-- in this case, first becoming Super Mario and not sustaining a hit-- involved in achieving it.

Ingeniously, **Super Mario Bros.** provides a reward for achieving the ultimate powered-up state that a game of four-square does not: only the use of fireballs can defeat each of the Bowsers at the end of each castle. Regular or Super Mario can run under or leap over the boss, touching the axe that destroys the bridge, sending Bowser to a lava-y grave with no points awarded, but only Fiery Mario can hit him with fireballs until he expires, scoring eight thousand points and a feeling of satisfaction seldom rivaled to this day.

The desire to achieve this-- to defeat the boss with fireballs, and, indeed, to defeat every boss with fireballs-- invests the Fire Flower with incredible value. It becomes deathly important not to get hit, to remain Fiery Mario until you reach the castle. The sparsely-populated castle stages become especially unnerving as the player must make carefully-timed leaps over pits of flame-spewing lava and fiery chains-- one misstep and the chance to destroy the boss face-to-face is gone. At that point, the player must bide his time, evade, and sneak, killing the boss by cutting the bridge from beneath his feet while behind his back, a deed made more demoralizing by the fact that the player *was so close...!*

In the power-up hierarchy of **Super Mario Bros.**, the Fire Flower is clearly at the top. The Super Mushroom, by contrast, is less powerful. The increase in size is largely an aesthetic improvement: Super Mario runs just as fast, jumps just as high and just as far as his non-super counterpart. The ability to break bricks with your head notwithstanding, the only real difference between the two is that if regular Mario is touched by an enemy, he dies, while if Super Mario is touched by an enemy, he is merely reduced to his regular Mario size.

What's unusual about this extra hit-point, the only "power" imparted by this power-up (and a power it shares with the Fire Flower), is the unwillingness of players to use it. Players play *more* cautiously with the extra hit-point than they do without it. Psychologically and on its face, this is strange: why would a skateboarder, for example, skate more carefully with pads and a helmet and, implicitly, skate more recklessly when unprotected? It seems wonky at first, as if the player is investing more value in the mushroom than it really has.

Just as the value of the Fire Flower is inflated because of its relationship to the Super Mushroom, the value of the Super Mushroom can only be considered in relation to its more illustrious cousin. It is this relationship, in fact, that makes the Super Mushroom valuable at all.

Consider if the Super Mushroom and the Fire Flower had no relationship, if regular Mario could become Fiery Mario with no intermediary step. What, then, would be the use of the Super Mushroom at all? Would a player play more defensively as Super Mario if he could become Fiery Mario even while small? In fact, would he play defensively at all?

The key to the power-ups in **Super Mario Bros.** and the genius behind them is not in the actual power-ups themselves, but rather in the relationships between them: a carefully-balanced *system* that is part-and-parcel with the larger system of the game itself. Not every game can or should duplicate the tiered system in **Super Mario Bros.**, but many games could learn from its deceptive simplicity.

And that's not to disparage the power-ups themselves, which, as noted above, change the game play dynamic substantially. It is for this reason that one cannot consider the other items in **Super Mario Bros.**, the collectable coins and the one-up mushrooms, to be power-ups at all. While both do contribute to the player's well-being by adding an extra life to the player's stock, this does not really change the way the game is played

either in real terms (i.e., an increase in tangible power) or psychological terms (i.e., an increase in the feeling of power, and the different play-style resulting from said increase). The "before" game and the "after" game are identical.

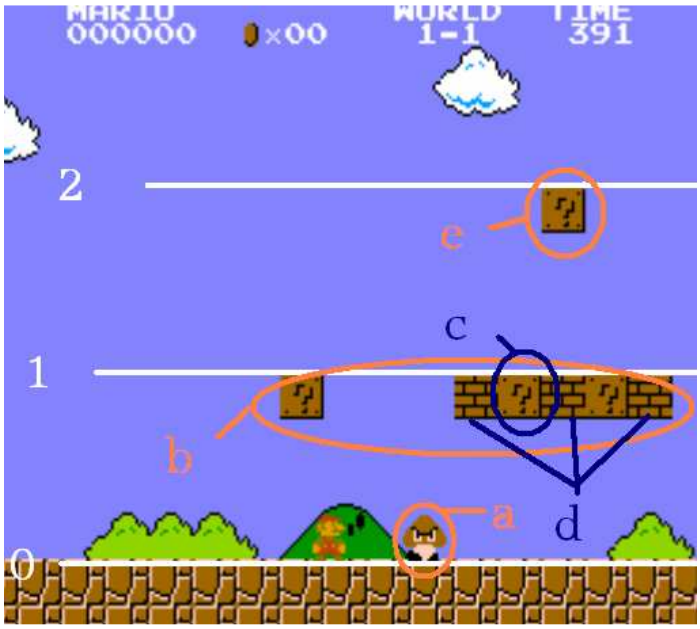
Lesson # 3: The Most Important Level

Dubbing any one level in a game the most important one is a bit of a deliberate misnomer. Just as the power-ups in the preceding discussion are all equally important-- the fire flower cannot really exist without the mushroom and the star, the star without the flower and the mushroom-- and just as the various elements in the game-- power-ups, controls, enemies, level design-- have to work and flow and fit together seamlessly on a systemic level, all of the game's various stages and challenges cannot exist without the other. There should be no weak links or padded fluff in vigorous game design.

With that fully in mind, however, the most important level in any video game will without exception be the first one. It is the first level that introduces us to the game's mechanics, world, style and feel. It is the first level that gives us an immediate sense of what is in store for us. It is the first level that makes us want to play the second, and the second the third, and so forth; this flow of desire begins with the very first stage, and if it does not generate the necessary momentum, then the player has no reason to play the game. (This is especially important on the amateur level; too often, as I'm sure we're all aware, players go into a freeware game looking for a reason to stop playing it, just as film festival selectors go into a submitted film looking for a reason to turn it off. It is vitally important for the independent artist not to give them that reason.)

The first level of a game, then, must contain the entire game in miniature without exhausting the game's content. Lucky for us, **Super Mario Bros.** pulls it off beautifully. I dare say stage 1-1 is the best first level design I have ever encountered and perhaps the best one I ever will.

When the level begins, the player is confronted with a flat side-scrolling landscape, devoid of enemies or objects. Scrolling forward, he comes to the first "real" screen of the game-- one of the most iconic and important screens in gaming history. Let's look at this screen together, carefully and deeply.



This screen not only introduces the first enemy, the goomba (a), but also the primary way of disposing of or avoiding it-- by jumping or stomping. It also introduces the first bricks and question-mark blocks (b), which encourage the player to be curious about their contents due to the emblem emblazoned upon them, and thus another jump-based action: namely, bonking the blocks with your head to yield coins and the game's first power-up (c), the Super Mushroom. Armed with that Super Mushroom, one can bust the bricks (d) to pieces.

The floating block situated on the second tier (e) seems a bit strange at first. It does not contain the power-up, nor does it "lead" anywhere like the high-tier blocks in the game's latter stages. Why did the game's designers put it there at all? Would the game be any better without it?

No, it wouldn't; that block actually has a vital role, perhaps the most vital role. Just as the first enemy teaches the player how to destroy enemies, just as the blocks on the first tier teach the player how to "bonk" out coins and power-ups, and just as the super mushroom is situated so that the player can instantly perceive its value-- it allows you to break up the brick blocks nestled around it-- that second tier block teaches the player that they can and should jump up onto the first tier blocks, that they have multiple paths available to them, that they should be curious and move boldly and investigate and experiment.

All this together, on this very first screen, sets out in no uncertain terms some of the most basic and important mechanics of the game. If the entire first level does contain the game in miniature, this first screen contains multitudes and does the brunt of the work.

Of course, that's not to say that the other screens aren't important; they certainly are. But that doesn't make this first action screen any less a marvel of compact exposition. The first power-up is placed near something that can demonstrate its power. The first enemy and the first non-enemy objects that the player can interact with are positioned together in a way that emphasizes the common action-- jumping-- which unites them both and forms the central idea of the game itself.

Immediately after that action-packed sequence (I killed the goomba! I got the mushroom! I can break bricks!) we encounter the game's first two pipes and the empty space between them, with another goomba on the other side.

Hmm. I wonder why they did *that*? Wouldn't it make more sense to have a goomba in between those first two pipes? I mean, that's about three-quarters of a screen where nothing's going on-- no power-ups, no coins, no enemies, nothing!

But there are, in fact, two very good reasons to have all that empty space. The first is the reason best summed up by the eloquent Gregg Tavares in his invaluable article, "Game Design Secrets of the Greggman"[10]:

> *All levels should be separated into action, rest, action, rest areas. One easy way to do this is the have 'blocking' graphics. Graphics that allow you to separate one situation from another. Like the pipes in* **Mario**. *The pipes in* **Mario** *not only separate one area from another in terms of play but also separate groups of enemies so that the player can't get too many enemies on the screen at once which would push the game engine to it's limits. (i.e., slow down)*
>
> *Separating areas like this helps you look at an area and gauge how hard it is in a smaller context than the entire level. Can the player make it thought this one area to the next rest spot. Longer areas are generally harder to survive so you can often get a progression through a level by making the next area a little longer than the previous area.*

As usual, that's sound advice from the Greggman, and it is of practical value both in terms of designing a game that players want to play and in terms of making it easier to troubleshoot. If you don't separate each stage into "mini-stages", it's harder to tell what part of a level isn't working when a level isn't working; you'll have to rethink or rework the entire thing. Whereas, if you separate it into chunks, you can look at a given chunk, see that it is not working, and redesign or remove it with the good parts of the level remaining intact.

But, as I said before, there is a second good reason for this use of negative space, and that's the same reason the game actually starts with one screen that's all negative space: it gives a sense of openness. Imagine if the game actually begun on that "first" screen we dissected above; as soon as the player pushes the start button, he's right next to the goomba and right under those first blocks. Besides being grossly unfair-- and that's one reason why the very first screen of a given level should be "empty", as it allows the player to prepare and to find their bearings-- it would also be kind of claustrophobic. The player would feel hedged-in.

And that, frankly, is the exact *opposite* of the vibe engendered by **Super Mario Bros.** Instead, it

[10] http://games.greggman.com/pages/game.htm

thrives on a feeling of openness, of expansiveness, of a whole world to bounce around in. By putting such considerable space between those pipes, that feeling of openness is emphasized. The entire fun, bouncy vibe of **Super Mario Bros.**, captured so endearingly by Koji Kondo's theme, *depends* on areas like this one.

Three goombas later, we come across a screen that contains not only the first of many bottomless pits but also the game's first two secrets and the first instance of a branching path. The player can choose to go down the pipe, thus skipping over much of the remaining stage, or he can proceed to the right. The savvy player who choose the latter path will also be able to grab the hidden one-up mushroom just underneath the spotted hill in the background.

But let's talk branching paths a moment, both in terms of the overall design of **Super Mario Bros.** and in terms of its specific function in this very first level. While the branching paths throughout the game still skew to a linear path-- castles and warp zones aside, all pipes will pop you out somewhere further along the line in the same level, and even warp zones move you along to a numerically-higher world, with Minus World, of course, being a notable exception-- they are emblematic of two of the most elusive but important qualities of the video game art form, player choice and player autonomy. That is, left-to-right linearity or not, the player can make meaningful choices-- in this case, to go down the pipe or continue over the pit-- and the player is in control of when he makes those choices. It is this second quality of player autonomy, of the player's ability to do what he wants when he wants to do it, that separates a video game from such dubious enterprises as the Choose-Your-Own-Adventure book and interactive movies.

Offering the player meaningful choices and offering the player meaningful ways to interact with the world around outside of the context of those larger choices are the key promises of game design that the use of branching paths in **Super Mario Bros.** helps to deliver on. More specifically, the use of the branching paths at this moment in the game establishes it as a regular feature of the game play and thus something the player should be looking out for. In today's games, in which the first level often foregrounds the process of teaching the player how the game works by making it a full-blown (and at times lengthy) tutorial, the use of branching paths during the first stage is often strictly verboten. This, in turn, helps to further set that tutorial level aside from the rest of the game; with **Super Mario Bros.**, however, the first level folds effortlessly into and foreshadows with gusto all that follows.

Indeed, after choosing to leap over that pit, the player is introduced to the first of two context-specific offensive options. The next two goombas we meet give us the context for the first; they're descending from some floating bricks situated above the player.

We've already learned from that very first action screen that we can hit blocks from below; what happens, then, if we do so when there are enemies walking on them? Aha! We've now become canny to a very important part of our repertoire, a move we shall find infinitely useful when facing the dreaded Hammer Bros.

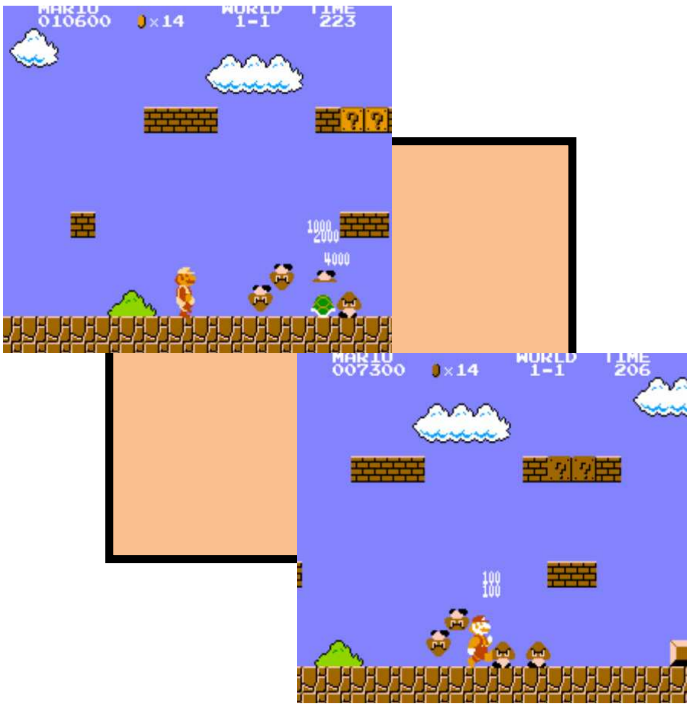
And consider how very important the location of this revelation is. If this was the very first group of blocks we had encountered, or if our very first goomba came at us from above like these ones do, the player might not of put two and two together as easily. Because the first action screen taught us to bonk things with our head, this one reinforces and expands upon that discovery. Also reinforced is the idea that we can and should jump up and move on the floating blocks; if the goombas can walk up there, why can't we?

It also provides us with another instance of a branching path, by offering us both a low road and a high one. It's not a particularly meaningful branching path; unlike the earlier instance of the secret-area pipe, this one doesn't cover much ground and there really isn't much difference between the two options. But, like the second-tier floating block and the use of empty space between the pipes we looked at before, it increases the *feeling* of freedom and reinforces the *idea* of branching paths.

Before we climb up on those blocks or leap over that next pit, however, there are two more things about this small and sparsely-populated little island that we should take note of. First and perhaps foremost, that question mark block upon which those goombas had traversed contains the game's first fire flower and thus the game's next-- and perhaps most powerful-- offensive ability.

Secondly, it is here that we find the game's very first checkpoint, just to the left of the next pit. *If you get far enough along before you die*, this tells the player, *you don't have to go all the way back to the beginning*. Besides saving the player a bit of grief, the use of checkpoints also help the game's designer to more clearly define the various "mini-stages" of a given level, and to more accurately determine their levels of difficulty. We shall see very shortly that the next "mini-stage" is the longest and most difficult part of World 1-1, featuring the level's highest concentration of enemies. Conversely, it also features the most choices in the level, as it not only offers a number of things to climb onto, thus allowing you to avoid those enemies, but also-- with the introduction of the last two offensive options-- four possible ways to attack them.

The first of these new ways to attack is one of the simplest, the goofiest, and the most rewarding, in terms both tangible (points and extra lives) and intangible (fun). It's something that we mentioned by way of example in our previous discussion of the game's power-ups: the kicking of a koopa shell and the resulting doubling of points. This, too, is a jump-based action, and thus extremely intuitive. If the player jumps on that koopa troopa and sends him into his shell, he is tempted to jump on it again, thus sending it flying and knocking out whatever enemies are in view. To make full use of this maneuver, the player will have to chase after it, which in turns mean he will miss out on the items situated above him. One of those items is his last major offensive option and the last of the three power-ups: the Starman. Equipped with the Starman, all Mario has to do now is simply run into those enemies to kill them.



And so, there are four distinct ways, allowing four distinct styles of play, to dispose of the nine enemies in this section of the level. The player can (a) stomp on them, (b) fling fireballs at them, (c) kick the shell at six of them, or (d) use the temporary invulnerability offered by the Starman. Together with the previously-established (e) hitting

blocks with enemies from below and pit-jumping 101, we have by now learned nearly every option the game has made available to us. All in the first level, all very smooth and intuitive without unnecessarily calling attention to itself.

There is, of course, more to this first level-- another jump, the exit for the earlier secret-area pipe, and two more goombas await us before we reach the first stairway and flagpole which signify the end of the first stage. But some of you may be wondering, apropos that secret-area pipe which skips over so much of the level and the fact that if you exercise one offensive option (for example, kicking the koopa shell) you might eliminate others (such as the Starman), if that negates my claim that it's all there in the first level. And it is true that there is no way to do, experience, and learn everything during the same play-through.

But, in a way few others do, World 1-1 delivers on the promise of the first level as it really does contain the entire game in miniature. Not merely in the ways we've discussed above, but in a perhaps more important though less concrete way, in that it embodies the spirit of the game. It gives the player freedom and encourage the player to explore. You will not learn everything about the first level the first time you play it, but the next time you play it there's a very really chance that you'll discover something new.

That thrill of discovery and the implicit freedom to explore and thus discover is part of what **Super Mario Bros.** is all about.

Lesson # 4: New Content

There are thirty-one other stages in **Super Mario Bros.**, and over half of them introduce *new content*-- that is, a new type of enemy, obstacle, or challenge. While the first level should contain the game in miniature, that doesn't mean it should exhaust all the game's ideas. Indeed, creating viable new content is essential for any winnable game, and so the spacing of that content is one of the trickiest decisions a game designer has to make. On the one hand, you want to save enough of your ideas for the end of the game so that it remains interesting; on the other hand, if there aren't enough ideas in the game's early stages, no one will keep playing long enough to get to "the good stuff". The solution, of course, is to have more than enough good ideas to go around, but it's not quite that simple; each idea must still fit with every other idea, must in some way function as an extrapolation from the game's center.

That's something that's easier said than done, but it's something that **Super Mario Bros.** more than pulls off. And so, let's take a look-- level-by-level-- at the distribution of new content in the game's remaining thirty-one levels. We'll also be zeroing in on the special non-mimetic themes that some stages are centered around, and the ways in which certain ideas are expanded upon to ramp up the difficulty. After this more general over-view, we'll take a look at the evolution and spacing of certain types of new content and the design decisions that make them tick.

1-2

This stage, the first "underground" level, introduces the first really tricky jumps in the game. What's really worth noting, however, is that these jumps pose no real threat to the player. He's not jumping over bottomless pits; if he falls, he simply gets back up again.

In this way, the game prepares the player not only for the tricky jumps later in the game, but also the elevator-platform jumps later in the same level.

This is also the first claustrophobic level. This is hinted at in that tricky jumping section by the narrow spaces between the pillars, but is at its most overt in the congested high-traffic area just a couple of screens over.

Even the secret coin area in this level is tighter and more claustrophobic than the one in 1-1. In fact, it is that comparison to 1-1 that gives 1-2 its claustrophobic aura and its heightened sense of danger. And it is precisely because the player feels so hedged in that he seeks-- and may find-- a way out. He already knows that Super or Fiery Mario can bust up bricks with his head; this knowledge, along with a need to get out of there, might compel him to bust up some bricks in the ceiling, thus allowing him to take the safest of the game's "high roads".

This, in turn, may lead him to jump from the last elevator platform to the bricks situated above the exit pipe, which will take him to the Warp Zone. The 1-2 Warp Zone is one of the game's worst-kept secrets, but that's not because everybody's told everybody about it; instead, it's one of those secrets that are very easy to discover on your own. Indeed, the game practically compels you through 1-2's tighter and deadlier level design to figure it out-- one discovery leading to another.

1-2 also features the first *red* Koopa Troopas and the first piranha plants, the latter of which adds a degree of timing to the player's jump from pipe-to-pipe. This, in turn, builds on the tricky jumps that began the level to prepare you for the elevator jumps that await you at its conclusion.

1-3

The next stage builds on the tricky jump motifs of the previous level; this time, it's all tricky jumps and all of them are potentially dangerous. It's also far more linear than the first two levels. While there are high roads and low roads in abundance, there are no pipes or secret areas to discover.

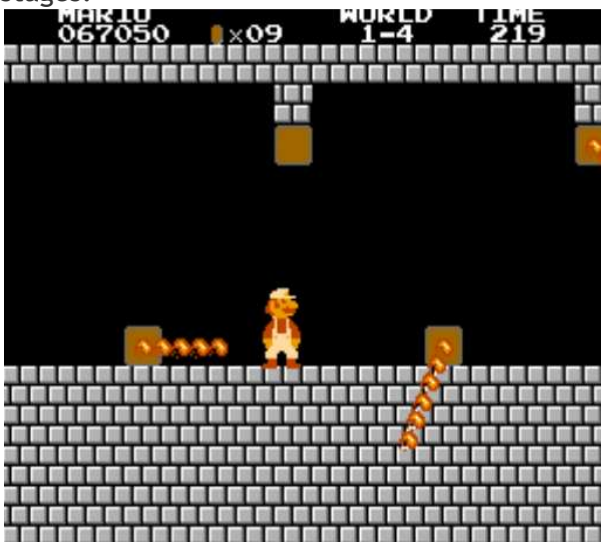
It is, however, much more "open" than the preceding level, so much so that it encourages the player to bounce along quickly from cliff to cliff, platform to platform. This, in turn, makes those tricky jumps all the trickier, as a player who is moving faster is also more likely to miss.

Compounding the situation is the presence of horizontally-moving platforms. While the vertically-moving elevator platforms gave us a small margin of error, these offer none; you can't simply jump a little later and end up on a platform a little lower. If you don't jump at just the right moment, you won't make the jump at all.

In keeping with the falling-from-great-heights motif, this stage introduces the Red Paratroopa.

1-4

And now we arrive at the first castle level, and it is a whole new ball game. There are no enemies here-- at least none of the stompable and flammable variety that we have, through-out the last three stages, come to use as a definition of "enemy"-- but rather the indestructible firebar and Bowser's own fire-breath, which test those tricky jump skills we've been developing over the last two stages.



The shortness of the stage, the presence of only one power-up and the lack of destructible enemies all conspire to create a certain austerity, a certain focus: this is level is all about Mario and Bowser, about the player and the game's first boss.

Some Thoughts on World One

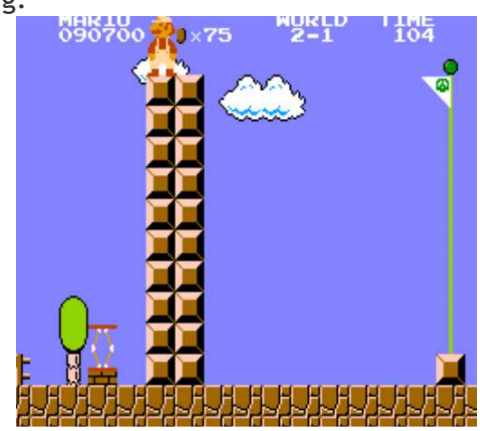
While 1-1 introduces the player to the general game play of **Super Mario Bros.**, World One as a whole sets the standard for the seven words that follow it. The first stage will always be an over-ground platforming stage with a certain degree of openness. The second stage (with a few exceptions) takes us underground or underwater (more on that in 2-2). The third stage is always about speed and jumping and is always aggressively linear; the final stage is always a castle. This allows the player in many cases to prepare themselves for the types of challenges they face. It also makes it easier for us to chart the progression of certain challenges common to certain types of stages.

2-1

After all that claustrophobia and those precision jumps, it's nice to get back out in the open with a clean and simple bit of platforming. It's not all roses and soda pop, however; this level introduces us to the most persnickety of all the turtle enemies, the Green Paratroopa. Unlike its red cousin, who we met in 1-3, this fellow bounds around from right to left and back again, making him a hazard whether you're on the ground or in the air.

On the brighter side, this level also introduces the first vine or beanstalk, which takes us to the sidescrolling "coin heaven" bonus world.

At the end of the stage, we are not particularly pleased to make the acquaintance of the trampoline platform. Depending on your timing, this will either propel you upwards or simply cause you to bounce. Later, this will be a source of considerable consternation. At this moment though, the trampoline is introduced in a "safe" context; there are no pits nearby for us to fall into. Recall how the first set of tricky jumps in the game, in 1-2, also took place in a "safe" context. By allowing the player to get used to a type of challenge without penalizing them for failure, we make the introduction of those challenges less daunting.



2-2

In 1-2 we went underground; in 2-2, we head under water. Our normal store of enemies are nowhere to be seen, instead replaced with the octopus-like Blooper and the fishy Cheep-Cheep. Gone as well are power-ups to find and bricks to break, secret areas to access and pits to jump over. Indeed, once under water we find even our jumping ability, the cornerstone upon which the game turns, conspicuously absent. We don't stomp or leap but instead fling fireballs and swim. And should we not have a fire flower, then we do a *lot* of swimming, as fast as we can.

A question might be begged: does this violate that number one rule of systemic coherence? When playing the underwater levels of **Super Mario Bros.**, are we still playing the same game? Or do I make too much of this-- is systemic coherence not that big of a deal after all? Have I been too hard on skateboarding and ice levels?

These aren't questions that I necessarily have answers for. And it is on just such an occasion that I implore you, oh thoughtful reader, to weigh in by sending a letter or even an article to this fine publication.

2-3

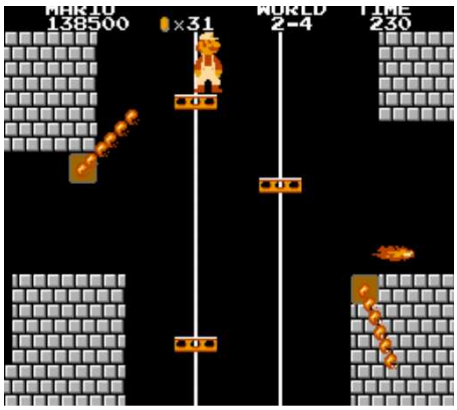
The jumps in 2-3 might be less frequent than those in 1-3, but they are no less tricky. This is thanks in no small part to the presence of a second kind of Cheep-Cheep, now with a decidedly more avian bent. The little critters leap from below in a never-ending supply-- also marking the first instance of an uncountable enemy.

This level is even more aggressively linear than most of the dash-threes; there is no high road or low road but just one road laid out from left to right. I imagine this is to accommodate the particular sort of challenge posed by the wave-after-wave of flying fish; can you imagine trying to navigate the sort of landscape in 1-3 *while* contending with the soaring Cheep-Cheeps?

More than its World One counterpart, 2-3 tasks the player with moving *but fast* and the linear layout cheers them on.

2-4

This castle ups the ante in a number of ways: there are more jumps and more firebars; the Podoboo-- a sort of indestructible fireball that leaps upwards from the lava below-- is introduced; elevator platforms, far tinier than any we had to contend with in World One, require extreme precision. And then when we get to Bowser, we find that it won't be quite so easy to leap over him this time around, as the top of his chamber is lined with smashable bricks.



3-1

Perhaps the most notorious and frustrating of the game's regular enemies, the Hammer Bros., make their first appearance in this night-time stage. With rapid-fire projectile hammers, the ability to leap up onto a platform from directly below it and vice-versa, as well as their mostly stationary behaviour pattern-- unlike the goombas, they won't simply walk off the screen: it's up to you to defeat or run past them-- they are formidable and dreaded opponents.

That's the only real major introduction in this stage; mostly, it's content to ratchet up the difficulty in other ways by confronting us with groups of Green Paratroopas and bringing back the trampoline from 2-1 with the safety net removed. If you don't time this jump correctly, you won't propel onto the blocks but rather bounce right into the bottomless pit.

3-2

This stage cools it as far as new content is concerned but finds an interesting twist on that content that's already been introduced. Previously in our time together, we've looked at the domino scoring mechanism that doubles the score with each successive enemy hit by a kicked shell. This stage is actually built around that principle; there are five separate "clusters" of enemies, each of which is begun with a type of Koopa Troopa and each of which, thusly, can be wiped out with a little deft jumping. It's a fun and breezy sort of level-- a nice breather between the trauma of the Hammer Bros. and the high cliff hijinks that await us in the next.

3-3

By now, we know (and love?) the vertically-moving elevator platforms and tolerate their less-prevalent horizontal cousins. This stage complicates matters further with the introduction of the scale or pulley-platforms. Stand on one side and it'll go down, causing the other to move upwards. Stay on too long and your platform will fall the rope, never mind the fact that you wouldn't be able to reach the other platform.

3-4

In keeping with World Three's slowdown in new content-- all we've seen are the Hammer Bros. and those pulley-platforms-- its final level doesn't bring much new to the table either. There are more firebars, more pits, more Podoboos, and a trickier boss chamber. But this castle doesn't really offer the same level of challenge or ingenuity that we saw in 2-4.

Some Thoughts on World Three

I think the stage that's most indicative of World Three isn't 3-1 or 3-3, in which new content is introduced, but 3-2, which simply provides a clever itineration of familiar elements. The Hammer Bros. and the pulley-platforms aside, 3-1 and 3-3 are about the same thing: taking what we've already learned about the game and making it harder. This whole world is a bit of a breather-- a not particularly restful respite between two worlds absolutely packed with new content.

4-1

Secret pipes aside, this level is far more "open" and less complicated than most other dash-ones. This is for much the same reason that 2-3 is more "flat" than the other dash-threes; like that level, this one introduces another supply of potentially limitless enemies in the form of the Spinies. The Spinies are unique among non-castle enemies in that they're the only *mobile* ones that absolutely cannot be stomped upon; doing so will cost Mario his life or, at the very least, his powered-up status.

They can only be felled by fireballs or starmen, and just as in 2-2, if you aren't equipped with those power-ups, you have no choice but to avoid them. (Perhaps this is one argument for 2-2 after all; it prepares you for 4-1.)

Also introduced (naturally) is Lakitu, who tosses the Spinies at you from above. Unlike the Cheep-Cheeps, the Spinies aren't actually limitless; stomping on or hitting Lakitu with a fireball will stop him, at least temporarily, from following you around and tossing the Spinies to the ground.

4-2

This stage takes the claustrophobia and tricky jumps of 1-2 to an extreme, confronting the player right from the get-go with narrow platforms separated by huge pits and a scrunched-up little corridor that's full of enemies with hardly any room to jump on them.

(Thank goodness for fireballs.)

This sense of claustrophobia-- heightened by the extreme openness of the preceding stage-- even extends to the discovery of the level's warp zone, which requires the player to hit certain invisible blocks but ignore others.

As far as new content is concerned, this level also confronts us with the Buzzy Beetle, the first *fireproof* enemy in the game. He can't be killed by jumping, either; like the Koopa Troopas, this only sends him into his shell. Doing so, especially in the tight spots in which we first encounter the species, is highly dangerous.

While 2-2 introduced us to a whole new type of game play, in which the regular rules don't apply, 4-1 and 4-2 show us exceptions to those rules within the context of normal game play. You can stomp on enemies, except for the Spinies and the piranha plant; you can kill enemies with fireballs, except for the beetle.

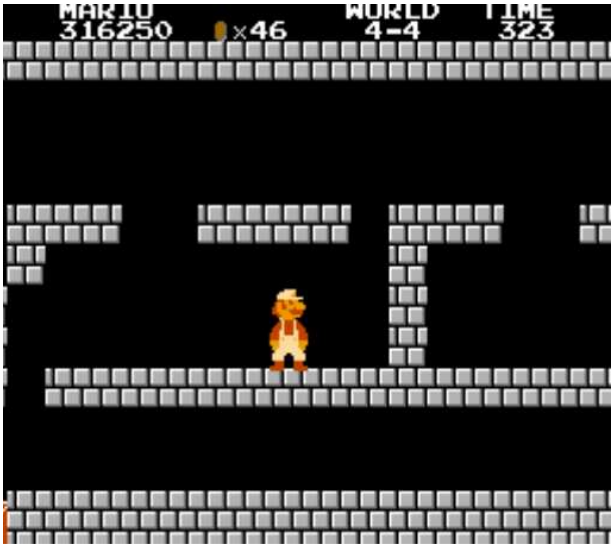
This is a dangerous new world we're in, and it's only going to get deadlier...

4-3

4-3 throws more pulley-platforms at us, and also makes them more difficult; we now have to stay on the first long enough to bring the second one up but not so long where it ends up out of our reach. This isn't quite as difficult as it sounds: underneath each set of platforms is a mushroom cliff-face. This is another instance of gradually increasing the difficulty with a safety net in place; the next time we see the pulley-platforms, we shall not be so blessed.

4-4

The World Four castle throws something new our way but it doesn't take the form of a new enemy or type of platform. Instead, we now encounter the first and easiest of the "maze" castles. The game confronts us with two to three different tiers of platforms; if we choose the wrong direction, we are placed back at the maze's beginning. Trial and error (or a quick google search) will allow us to arrive at the conclusion.



We'll be talking about the maze castles, and how this might be one of the game's few real flaws, later on. For now, let's move on to the second half of the game. That second half has far less new content than the first; we'll be talking about that later as well.

5-1

World Five introduces us to the final new enemy type, the Bullet Bill. Like Buzzy Beetle before him, Bill is invulnerable to your fireballs, though he is quite vulnerable to your stomp; like the Spinies and Cheep-Cheeps, he can respawn indefinitely, flinging at you from his cannon.

This stage also gives us a lone Hammer Brother *sans* platform; this one guards a block staircase. There's no way to sneak underneath him this time.

5-2

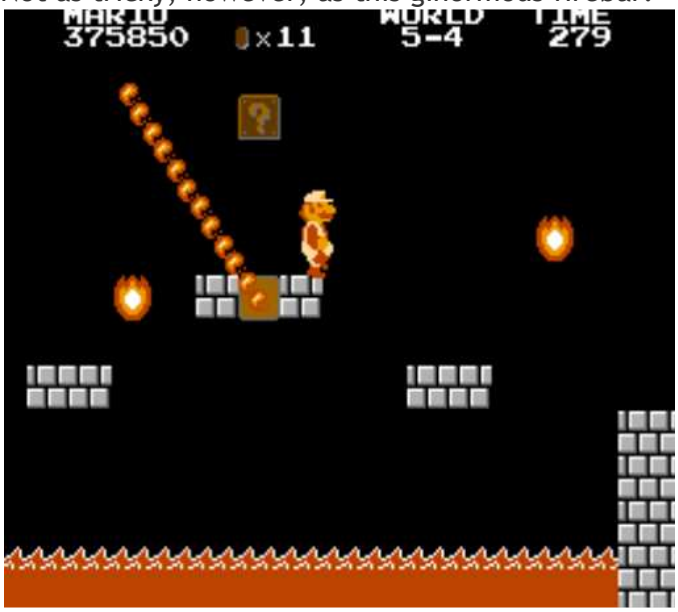
5-2 doesn't introduce new content so much as it introduces old content in a new context-- albeit one that's somewhat hidden, as it is contained in one of the bonus areas. Go down the right pipe and you'll find yourself facing those pesky elevator platforms-- *underwater*. Now, instead of trying to jump on them, you're trying to swim past them before they push you down into a bottomless pit.

5-3

Bullet Bill gets an upgrade in this level; now, instead of flinging at you from a cannon, it flings at you from the right side of the screen. He is no longer restricted to one predetermined horizontal plane, but is rather propelled at you along different horizontal planes depending on your position. This can get tricky.

5-4

Not as tricky, however, as this ginormous firebar.



Passing this bar is somewhat significant. Not only is a testament to a player's skill, but in crossing this line, we cross the line between the introduction of new content (Worlds One through Five) and the testing of the old (Six through Eight). We're in a whole new ballgame now, and from this point on, the game is significantly harder.

Lesson # 5: Ramping Up the Difficulty

6-1

This level takes the concept of 4-1-- there are a lot of Spinies, so let's make it as open as possible-- and inverts it: there are a lot of Spinies, so let's make it as dense and precarious as possible. This stage is full of tight spaces and difficult maneuvers, with the Spinies falling onto and walking off multiple levels.

6-2

One thing we haven't made a whole lot of in our discussion of the game thus far is the presence of a timer, typically set at 399 seconds (and counting). Generally, that's because the game gives us more than enough time to poke around and experiment.

But in this stage, which contains four different bonus areas-- one of which, an underwater segment similar to the area in 5-2, makes a dubious case for a "bonus" area-- it's quite possible for the player to run out of time if he isn't careful. If the previous five worlds encourage exploration and reward curiosity, 6-2 discourages it. Like 6-1, it turns the common sense established previously in the game on its ear.

6-3

Memory can be a tricky thing. Some months back, I wrote an article for *Game Maker Technology Magazine* in which I explained my long-held prejudice against ice levels as an arbitrary way to ramp up the difficulty. Instead of making things more difficult through actual level design, game designers simply take a level without difficulty, turn down the friction, and viola!, ice level: instant difficulty, no brains required.

In that article, I made a passing reference to **Super Mario Bros.** and the ice level in 6-3. So imagine how utterly stupid I felt when I played through that level again and realized that Mario doesn't slip-and-slide any more there than anywhere else. The friction is exactly the same. Though the platforms are white and snowy, this is not an ice level as I understand and despise it.

So why was it that I was so convinced that 6-3 was that type of ice level? I think it has to do with the way the different obstacles and elements are arranged. The tiny elevator platforms, the springs, the Bullet Bills, and the introduction of the first falling platforms (the very, very, very last piece of new content in the game) all emphasize-- and conspire against-- quick and accurate jumping. Because of this, it *feels* like things are more slippery, like we're moving faster than we want to; it *feels* like an ice level even as it plays fair in ways an ice level normally doesn't.

6-4

Far from introducing new content, the World Six castle unabashedly recycles the oldest stuff in the book-- or, at least, the game, for 6-4 is actually 1-4 with some Podoboos and extra firebars. This is not actually the first or only time this happens in the game. 5-3 and 5-4 recycle 1-3 and 2-4, while 7-2 and 7-3 take their cues from 2-2 and 2-3.

7-1

I am firmly of the opinion that the 7-1 is the work of a Bullet Bill fan club. Multiple cannons piled one on top of another and firing from various directions make this one of the most complicated levels.

7-2 and 7-3

As mentioned before, these levels are taken almost verbatim from 2-2 and 2-3. 7-2 adds over twice as many Bloopers, while 7-3 adds some Koopa Troopas to the Cheep-Cheep feeding frenzy.

7-4

The World Seven castle is light on obstacles but heavy on head-scratching, as it is in my opinion the least intuitive and therefore perhaps the most difficult of the three maze castles. The possibility that you will run out of time attempting to navigate its labyrinthine sequence of platforms is very real indeed.

8-1

The last of the game's eight worlds is the most unforgiving. While the non-castle levels of the previous seven worlds, regardless of their difficulty, provide the player with a checkpoint, World Eight does no such thing. This means that in order to clear a stage, you must be able to do so in one-go; you can't lose a life halfway through and start closer to the end.

Since this happens in the game's final world, you can imagine that it can, at times, become very frustrating. To say that some of the jumps in this extremely long level-- the longest stage in the entire game-- are precarious is to make a gross understatement.

8-2

The jumps in the dash-two stage, however, are perhaps twice as irritating.

Remember: if you miss this jump, you have to go back to the beginning of the level and try again. And again. And again.

8-3

The penultimate stage confronts us with multiple ground-level Hammer Bros. with some Bullet Bills thrown in for good measure. Quickly jumping over one flurry of hammers after another tests the player's timing and precision in a way that's at once more forgiving and yet more nerve-wracking than the mile-long jumps of the last two stages.

8-4

I said apropos 1-1 that the first level of a game should contain the entire game in miniature. The same is often true of a game's final level. While the first stage introduces the various actions the player can perform to meet the game's challenges, in the final stage those challenges are calculated to call upon all of the player's skills. Those that don't -- I'm thinking of the last level of **Kid Icarus**, in which the vertically-scrolling platform shooter becomes a shmup, or the final level of the otherwise magnificent **M. C. Kids**, in which only the skills you've learned in the last two or three levels come into play-- are often unsatisfactory.

But 8-4 of **Super Mario Bros.** is anything but that. This final itineration of the maze challenge is arguably the most enjoyable, as it revolves not around an abstract concept such as "which level of blocks am I supposed to jump on when?" but in something concrete and central to the game-- transport via green pipes.

It's the only castle to feature non-castle enemies-- goombas, Hammer Bros., beetles, plants, and Bloopers-- and the only castle to feature, wonder of wonders, firebars underwater! 8-4 calls on your ability to jump, to stomp, to use fireballs (if you've got 'em), to swim, to figure out where invisible blocks might be located: to think in the blink of an eye. While it might not possess the *je ne sais quoi* of more refined Last Levels, such as the ingenious Wiley stages in **Mega Man II**, it does manage to provide a satisfying conclusion to the first great console platformer.

Having gone through the game in general one level at a time, the time has come to look more closely at what that process has to teach us.

Lesson # 6: Enemies

The game's final new enemy type is introduced in 5-1, the game's seventeenth stage; he is also, by some counts, the seventeenth distinct enemy type. I count him as the eighteenth, as I count the underwater Cheep-Cheeps separately from their flying cousins, as both have different behaviours and weaknesses. The chart reprinted below (courtesy of **themushroomkingdom.net**), counts Bill as number seventeen and provides a level-by-level breakdown of enemy populations. As they explain on their site, the presence of a "Y" indicates "that the enemy is present, but there will be varying amounts due to game design. For example, the number of Cheep-Cheeps, Bullet Bills, and Lakitus (+ Spiny Eggs +Spinys) will vary depending on how quickly you finish the level."

World																	
1-1	16	1															
1-2	14	3	4	1													
1-3	3			3	2												
1-4						7	1										
2-1	16	6	8					3									
2-2			1						6	Y							
2-3										Y							
2-4						6	1				2						
3-1	14	7	5					5				2					
3-2	15	19	1					1									
3-3	1			5	1												
3-4						9	1				6						
4-1			4										Y	Y	Y		
4-2	3	6	8													3	
4-3				5	1												
4-4			1			5	1				1						
5-1	21	6	4					4									Y
5-2	4	1	3	1				5	3	Y		4				3	Y
5-3	3			3	2												Y
5-4						11	1				6						
6-1			1										Y	Y	Y		
6-2	3	1	25					2	3	Y						4	
6-3																	Y
6-4						11	1				3						
7-1		1	5					4				4				1	Y
7-2			1						13	Y							
7-3		1		3				3		Y							
7-4						1	1				2						
8-1	24	16	10					3								4	
8-2	2		4					12					Y	Y	Y	4	Y
8-3		1	3					2				8					Y
8-4	3		14			5	1	4	3	Y	1	1				2	
Totals																	
	142	69	102	21	6	55	8	48	28		21	19				21	

www.themushroomkingdom.net goes beyond the scope of a mere "fan site", or the various "shrines" that litter the internet, by providing a compendium of useful and interesting information about the Mario games. I'm not talking about tips, tricks, secrets, and bugs-- though they do provide an exhaustive and fascinating list of those-- but about such technical facts as the height and stride of Mario's various leaps which garner a deeper appreciation of the game on a systemic level, and about historical information that puts the game in context.

It really is a terrific and invaluable resource that I drew on many times during the composition of this piece. I recommend it highly to casual Mario fans and serious students alike.

From the data on this extra-nifty chart arise a few equally-nifty rules of thumb about the distribution of enemies. For example, the first thing that strikes me after giving this chart a cursory glance is how seldom Lakitu and his spinies make an appearance: 4-1, 6-1, and 8-2. Surely such a formidable group of enemies should crop up more often; after all, Bullet Bill is introduced one full world after Lakitu (in 5-1) and yet he still manages to make an appearance in seven levels! So what gives?

What gives is that Lakitu and those stages in which he appears are *special*. His modus operandi-- following the player around through-out the course of the entire level and tossing spiny eggs to the ground-- so completely dominates the level in a way that no other enemy does. The Hammer Bros. might be tougher, and those Bullet Bills might be pesky, but none of them control as much space as Lakitu does in the way in which he does. The levels that Bullet Bill and the Hammer Bros. appear in are levels *with* those enemies; the Lakitu stages, on the other hand, are *his* stages: they belong to him.

Those stages are especially memorable because they (and their master) are so unique. If Lakitu appeared in, say, a dozen-odd stages, those encounters would be far less memorable and enjoyable. It would become boring: just another Lakitu stage. But by only using him three times, the game's creators ensure that each time *counts*. That's the power of **enemy exclusivity.**

Too many game designers, on both the commercial and freeware level, forget this concept and its corollary, **level-specific challenges.** Let us digress slightly to another game that your humble editor holds in high-esteem, **Mega Man II.** That game (and, to a lesser degree, every game in that franchise) also makes a great deal of use of enemy exclusivity and level-specific challenges. It is only in Air Man's stage that the player will find fan-men and cloud-riders; only Quick Man's stage pits you against those one-hit death-tubes; only Wood Man's level pits you against three fire-breathing dog mini-bosses. If every level had featured a fire-breathing dog mini-boss, they would cease to be memorable or interesting. Instead, they'd likely be irritating: oh, great, *another* dog boss.

Enemy exclusivity and level-specific challenges.

Familiarity breeds contempt, and variety is the spice of life; it's better to have a great enemy or challenge appear only once or twice than to diminish that greatness by overusing it. That's especially true if the enemy is an especially difficult one, like Lakitu or the Hammer Bros.

And while we're on the subject of variety, take a moment to look at our collection of enemies: no two of them are exactly alike! And that's as it should be; you should **never** confront the player with two different sprites that, for all intents and purposes, represent exactly the same set of behaviours and weaknesses. Every enemy should be different and every sprite must tell.[11]

"But wait a second," some of you might be saying. "Is there really that much difference between a green koopa troopa and a red one, or between the green and red paratroopas, for that matter? And how about a goomba and a koopa troopa-- don't they act the same way, aren't they vulnerable to the same attacks?

It's true that both a goomba and a koopa troopa will walk from the right of the screen to the left, and that if they bump into a pipe or another enemy, they'll walk from the left to the right. And it's also true that both enemy types are vulnerable to all the attacks in the player's arsenal.

But they are vulnerable *in different ways* to the player's stomp attack. While a good hard stomp will destroy the goomba, it will only send the koopa troopa squirreling into its shell. Another stomp or a kick will send that shell flying at whatever enemies are unlucky enough to be in its

[11] The reader is directed to "Sprite Substance", earlier in this same issue.

path. That, to my mind, is a remarkable and important difference in enemy behaviours. And that difference is indicated plainly to the player by the presence of the turtle's shell.

The differences between the green and red koopa troopa enemy types is perhaps more subtle; the green koopa troopa, when faced with a cliff, will, like the goomba before him, walk right off of it. The smarter red koopa troopa, on the other hand, will do an about-face. This doesn't necessarily make him any more deadly than his green cousin, but the difference *does* change the way the player approaches them.

Confronted with a green koopa troopa on the other side of a pit, the savvy player might leap over both and deftly avoid any entanglement; try that with a red one, however, and you're likely to be attacked from behind. It would be better in that situation to either wait for him to make his about-face or to time your jump so that you send him into his shell.

The paratroopa wings clearly differentiate them from their earthbound brethren, while the difference between the green and red paratroopas are far more striking. The red koopa troopa, remember, patrols from right-to-left and back again, controlling space horizontally; the red paratroopa, on the other hand, controls space vertically by hovering in place. In this game, that's frankly not very threatening-- even Lakitu's control of vertical space is dependent on his ability to also move horizontally, an ability the red paratroopa lacks-- which is probably why the red paratroopa only appears six times in the entire game.

Far more prevalent (and dangerous) is the green paratroopa. Like the green koopa troopa, he only move from right-to-left unless he bumps into something. If he encounters a pit, he's merrily jump into it. But unlike the red paratroopa, he controls space both vertically (he jumps up and down) and horizontally (as he moves left to right), which makes him considerably more difficult to avoid. That's a world of difference.

There are, in fact, two major points of differentiation that set each enemy apart from the other. One, mentioned in our example of the goomba and the koopa troopa, is an enemy's vulnerability to the player's various types of attacks. The second is the "deadliness" of the enemy, and this is not measured numerically (as every attack from every enemy will do the same amount of damage) but rather, and more tellingly, in the way that the enemy controls space.

In fact, **controlling space** (as David Sirlin will tell you) is what action games (and some strategy games, for that matter) are all about, and your enemy design should incorporate some thinking about how a given enemy moves and thus controls space.[12] Two different enemies that control space in two different ways are far more formidable than two different enemies who move and attack exactly the same but cause two different numerical subtractions from the player's pool of hit points. Those latter two enemies can be approached and defeated the same exact way, while the former require two different strategies on the part of the player. And if both of those enemies who control space in a unique way confront the player at the same time, then he has to come up with a third strategy.

Which brings us to another point: not only is **Super Mario Bros.** careful about when it introduces new enemy types and about how often it uses certain enemies, it's also careful not to throw too many enemies at the player at once. Not only are there seldom more than a handful of enemies on a given screen, but a quick glance at that handy chart indicates that only two of the levels, 5-2 and 8-4, contain ten out of the game's seventeen enemies-- the highest number. In fact, many levels only have three or four enemy types, resulting in an average over the course of the game's thirty-two levels of 4.125.

What all these numbers mean, of course, is simply this: **don't throw too much at the player.** Don't just throw all your enemy types together in one level (no, not even the last level) to make it more "difficult", as all you'll end up doing is making the game more frustrating and making your level too messy. Think carefully about how each enemy threatens the player, and come up with combinations that work together to increase that threat in a meaningful way.

For example, the Bullet Bills work well with Lakitu in 8-2 because while Lakitu is controlling space vertically while following the player horizontally, the Bullet Bills are control space horizontally; it's a criss-cross.

[12] See chart on following page.

DIFFERENTIATING TRAITS OF BASIC ENEMY TYPES IN *SUPER MARIO BROS.*

ENEMY	VULNERABILITIES	BEHAVIOURS
	Destroyed by all attacks.	Moves right-to-left; changes direction when colliding; falls into pits.
	Turned into shell by stomping or if attacked from below; destroyed by all other attacks.	Moves right-to-left; changes direction when colliding; falls into pits.
	Harms player if stomped; cannot be attacked from below; destroyed by all other attacks.	Pops up and down from pipe.
	Turned into shell by stomping or if attacked from below; destroyed by all other attacks.	Moves right-to-left; changes direction when colliding; changes direction at pits.
	Turned into Red Koopa Troopa by stomping; destroyed by all other attacks.	Flies up-and-down in a fixed space.
	Turned into Green Koopa Troopa by stomping; destroyed by all other attacks.	Bounces up-and-down from right-to-left; changes direction when colliding; falls into pits.
	Can only be destroyed by fireballs.	"Bloops" towards the player in an erratic fashion.
	Can only be destroyed by fireballs.	Swims from right-to-left on a slight angle.
	Destroyed by all attacks.	Leaps from below-screen either from the right to the left or left to the right, arcing near the player; respawns infinitely.
	Destroyed by all attacks.	Wavers right and left slightly; can leap up onto or below from floating blocks; throws dozens of hammers that arc upwards and fall a short distance from him.
	Destroyed by stomping, fire, or star; attacking from below and turtle shells not applicable.	Floats at the top of the screen, following the player through-out the level; throws spiny eggs downwards; will respawn at certain points if destroyed.
	Harms player if stomped; bounces if attacked from below; destroyed by all other attacks.	Moves right-to-left; changes direction when colliding; falls into pits.
	Turned into shell by stomping or if attacked from below; impervious to fire; destroyed by star and shell.	Moves right-to-left; changes direction when colliding; falls into pits.
	Impervious to fire; destroyed by all other attacks.	Moves right-to-left through all obstacles.

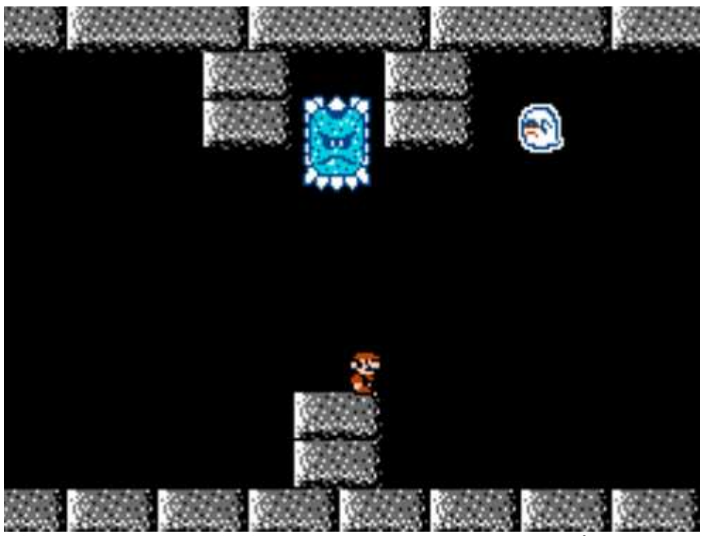
Each unique "trait" is given its own colour in my chart above. Note that even though several enemies share a given trait (the same vulnerability or the same space-controlling behaviour), no two enemies have the same exact traits in both categories; each enemy is unique. In good game design, every enemy (just like every level and every power-up) *counts*.

Lesson # 7: Obstacles

Designing and implementing obstacles is a bit easier than designing and pacing enemies for the simple reason that with most obstacles in a platformer, there is a sort of formula, a blueprint for the progression of difficulty. In the early levels, the various chasms that the player must leap across are smaller both in terms of distance and numerically. As the game proceeds, there are more-and-more pits, closer-and-closer together; the pits become wider-and-wider while the platforms used to cross over them become smaller-and-smaller. It's something that's so elegant and self-explanatory that you very seldom find anyone screwing it up. (I wish I could say the same thing regarding enemy design.)

Super Mario Bros., being a game that is at its essence about jumping and bouncing about, gets this right. (There are a couple of spots I have some reservations about, and I'll be voicing those very soon.) And all the other obstacles and challenges-- the elevator platforms, the scale platforms, the trampolines, the moving platforms, the platforms that drop out from underneath you, and even the firebars and podoboos that occupy the castle stages-- are all built around this progression; they're all devices that gradually tax the speed and accuracy with which you make your jumps. And they all work beautifully-- so much so, in fact, that they've become standards of the genre. You'd be hard-pressed to find a platformer that doesn't include some variation on ninety percent of these. And that fact carries with it some good and some bad.

First, the latter: because pretty much everyone and their grandmother has used this set of obstacles in their platformer, it's become very unlikely that you'll find anything *new* done with those elements. In many cases, this is the result of laziness; while the obstacles in **Super Mario Bros.** were designed from the inside outwards, working from the high concept (bouncing around) and extrapolating challenges from there, in many other platformers this simply isn't the case. The obstacles are just slapped on by rote, whether it's a particularly good fit for the game's high concept or not. And many platformers (let's be brutally honest here) don't have any high concept beyond "I want to make a platformer".[13] In such games, these obstacles seem derivative because the *entire game* is derivative.



The stone-faced thwomp from *Super Mario Bros. 3* provides a twist on the podoboo by threatening the player along a fixed vertical axis from above-- one of the few universal standards of platforming obstacles not contained in the original.

Even when a game designer tries to make these obstacles their own, there is really only so much you can do with a falling platform or a trampoline, or-- to extend our discussion temporarily beyond the auspices of **Super Mario Bros.**-- with a conveyer belt or grappling/swinging hooks or wall-jumps or rhythm blocks. And while there's always room for ingenuity, for new types of challenges, because these particular challenges work so well together and with that wider-pit smaller-platform faster-jumping formula, creating a genuinely new challenge (one that's just not a rehash of its predecessors) that meshes with the demands of the genre is a tall order indeed.

But, the promised upside of all this: these types of challenges *do* work well together and with the formula and they can provide lots of fun, even in those cases in which the execution is strictly run-of-the-mill. A game designer does not always have to reinvent the wheel, but if the game's high concept can give it a new spin-- well, then, that's something worth aiming for, yes?

[13] For more on the importance of the high concept, especially in a genre like the platformer, take a gander at the article entitled High Concept in our fourth issue, p. 3-9.

Aside from reinforcing the big idea that every design decision should flow out like blood from the pulsing heart of the game's high concept, the actual implementation of the obstacle design in **Super Mario Bros.** does give us something a little more concrete to grasp hold of. As mentioned before, many of the game's most devilish bits of platforming are foreshadowed early on with a **safety net.** The first trampoline is situated next to a wall; the next time we see a trampoline, it's right next to a bottomless pit. The gradually tighter jumps that unfold through-out the game are given a practice run in 1-2 without any pits to worry about.

While the push-and-pull of balancing a game's difficulty between too easy and too hard in order to arrive at "just-so" defies being reduced to an easy formula, I think a judicious use of safety nets when introducing a new type of obstacle helps tip the balance in the player's favour. Starting out safe and easy also gives you somewhere to go, something to build on and to gradually move away from.

Lesson # 8: Castles

The game's dash-four stages or "castles" are unique in that, more-so than any other "type" of stage (the underwater stages being a notable exception), they follow a particular formula. While many of the dash-three stages have a lot of things in common (precision jumping, speed, linearity), there is still a remarkable amount of variety in those levels-- 5-3 and 7-3 aside-- there are still a lot of big differences. But even when we take into account those dash-fours that introduce new content (podoboos, smaller elevator lifts, the maze castles), there is a startling uniformity to them, a certain appealing spartanness.

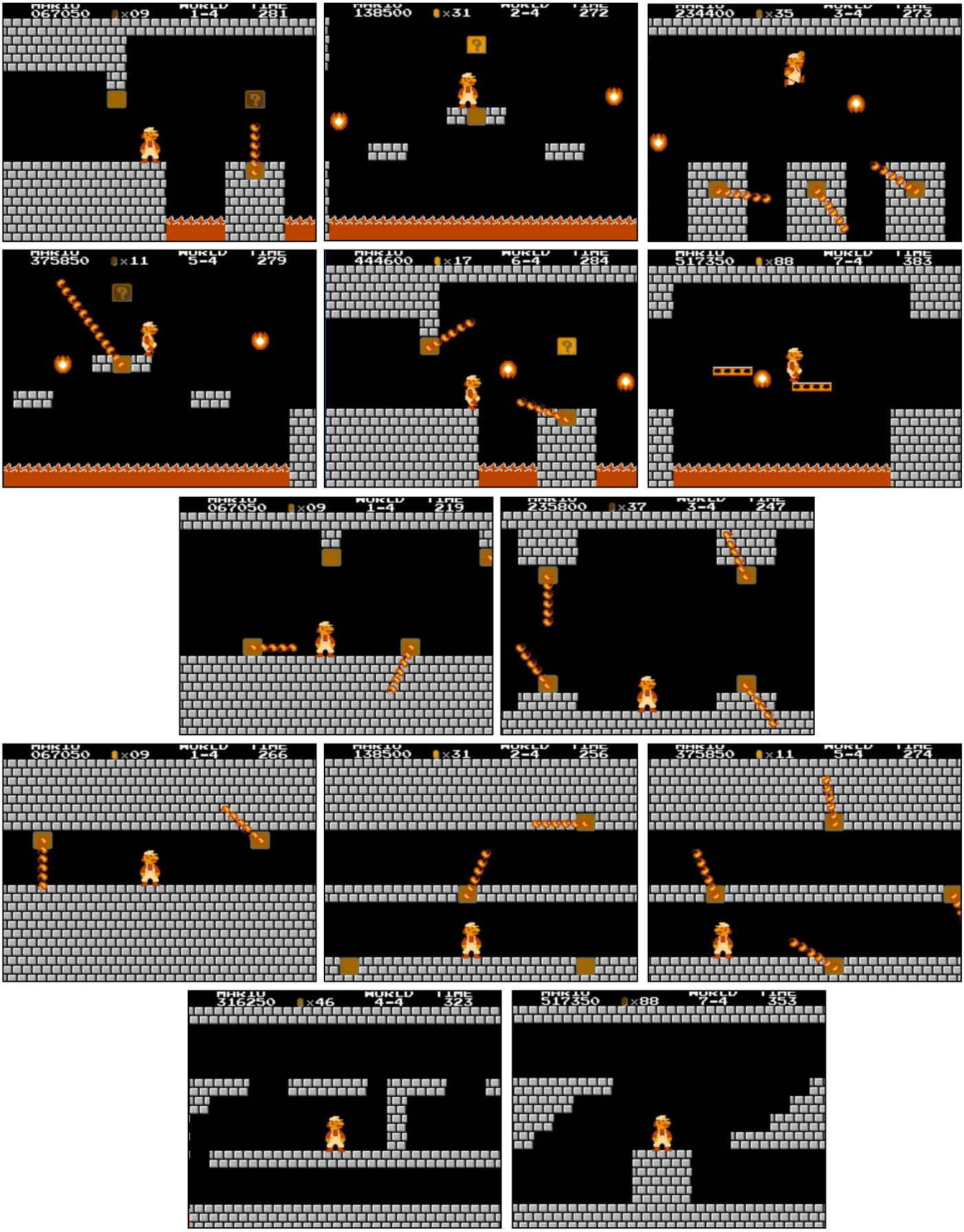
The level designs here are for the most part simple, challenging, and short. Their palette is very limited; setting aside 8-4's special status as the game's final level, the dash-fours feature no enemies and only one power-up. Coins are few and far between. It's all firebars and podoboos, lava pools and empty pits.

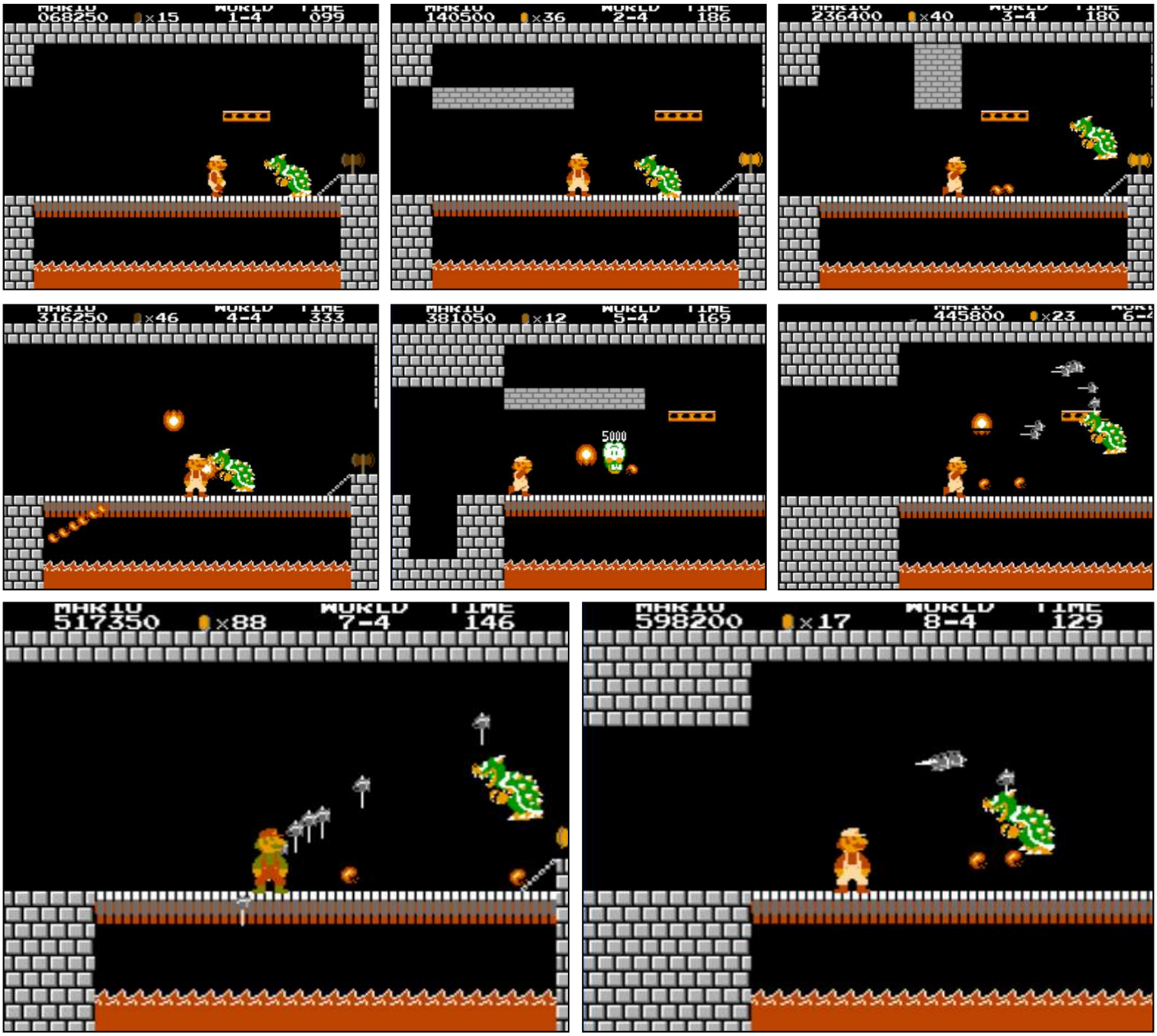
And yet despite or perhaps because of such a severely limited palette-- if the rest of the game is as alive and rambunctious as a Kurosawa film, the castles take their cue from the zen-like restraint of Ozu-- the dash-fours succeed in building towards the main event: the boss battle on a bridge with Bowser.[14]

More often than not, designers of platform games should embrace a little austerity. Hitting the player with one tricky challenge after another is more rewarding for both the player and the designer than asking them to navigate twenty different challenges at once. Too many games make it hard by making it busy. But making it busy doesn't make it *fun* and it doesn't make it *fair*. Or, to quote the man who wrote my fortune cookie the last time I ate Chinese: less is more.

[14] Try saying that ten times fast.

The limited palette of the dash-four levels has a nice unintended side effect for the scholars among us, as it allows us to see in stark visual terms the progression of difficulty from level-to-level and obstacle-to-obstacle.


MARIO 067050 ×09 WORLD 1-4 TIME 281
MARIO 138500 ×31 WORLD 2-4 TIME 272
MARIO 234400 ×35 WORLD 3-4 TIME 273
MARIO 375850 ×11 WORLD 5-4 TIME 279
MARIO 444600 ×17 WORLD 6-4 TIME 284
MARIO 517350 ×88 WORLD 7-4 TIME 383
MARIO 067050 ×09 WORLD 1-4 TIME 219
MARIO 235800 ×37 WORLD 3-4 TIME 247
MARIO 067050 ×09 WORLD 1-4 TIME 266
MARIO 138500 ×31 WORLD 2-4 TIME 256
MARIO 375850 ×11 WORLD 5-4 TIME 274
MARIO 316250 ×46 WORLD 4-4 TIME 323
MARIO 517350 ×88 WORLD 7-4 TIME 353



Lesson # 9: Flaws in the Crystal

As the preceding 12,000-plus words no doubt attest, your humble editor holds **Super Mario Bros.** in high esteem. And how can I not? It is a terrific and inexhaustible work that set the standard not only for the genre but for the art form in general. The ideas about game design that inform every choice its creators made and that we've talked about in the preceding pages have far-reaching implications for practically every type of game. And on a pure systemic level, it accomplishes so much with such economy that I have often referred to it, along with **Tetris**, as a perfect game.

But, looking at it seriously and as objectively as possible, I have to admit that dubbing it "perfect" (and thus beyond reproach) is a bit of hyperbole. It is, indeed, a great game, a classic; it is very nearly perfect in all the ways I have described. But there exists in this sparkling gem of a platformer two notable flaws.

And I'm not talking about the game's dozens of bugs and glitches ; it's not really a "buggy" game in that you really have to go looking for those glitches in order to find them. I'm talking about actual design decisions that, in retrospect, don't hold up under scrutiny.

First up, there is the extremely tricky jump in 8-2. This jump is *exactly* the length of Mario's running jump; the platform on the left of the jump is *exactly* the length needed to start running. In

order to make this jump, the player has to be *precisely* on the left-most edge of the platform. He then needs to run to the right and jump at *precisely* the right moment. If he jumps too soon, he won't clear the jump; if he waits too long, he won't jump at all but end up in the pit below.

Let me state this situation another way to ensure that the point sticks: the player must jump at *precisely* the right *fraction* of a second or he will die. And, because World Eight features no checkpoints, he will have to start over at the beginning of 8-2-- which, as you're no doubt aware, is no cakewalk.

It's certainly possible to make this jump. It's not a one-in-a-million kind of thing. It requires expert timing and expert skill. It is the ultimate test of the player's jumping ability; it is the final step of a progression that begins in 1-1. It's also too damn hard.

It's too unforgiving. And in a game that's all about bouncing around and making the choices you want to make, a challenge that can *only* be bested in *one specific way with no margin for error* is unforgivable.

I think I understand the thought process that's presumably behind this choice; it is, in fact, a natural result of the progression of difficulty at the heart of the genre. And I have no problem with difficulty, so long as it's not at the player's expense. (A checkpoint right before this jump would have been lovely.)

If I think I understand the thinking behind that decision, I'm ninety-nine percent certain that I *know* the thinking behind the game's other major flaw. In fact, we touched on that reasoning early on in this piece. Recall the story that Miyamoto told at the 2007 Game Developer's Conference about his decision to deprive the player of a sword at the outset of **Legend of Zelda**. "He theorized-- years before our internet age-- that players would have to talk to one another, share secrets, communicate, and figure stuff out together. "

This social aspect of gaming is a revolutionary idea. It is a principle that has been delivered on a thousand fold with online multiplayer and the do-it-yourself aesthetic of **Little Big Planet**. It is an idea that has clearly paid dividends. And even way back in the eighties, when many in the industry, still reeling from the Great Crash of '83, were unsure if video games even had a future in America, Miyamoto predicted and lay the groundwork for it in **Zelda** and **Super Mario Bros.**

Super Mario Bros. is full of secrets. Secret pipes, warp zones, hidden one-ups. Many of these things we stumble on by ourselves by accident but just as many were whispered to us by brothers, neighbors, friends, and fellow students. I myself was a lonely and shy little boy, and I had great difficulty connecting with people. I was often made the subject of ridicule and the victim of bullying. But those same bullies would gladly tell me about the "turtle tip" in 3-1, or the 30-lives code in **Contra**. Even if it didn't make me socially well-adjusted, Miyamoto's idea ensured I wasn't a complete hermit. I'm sure there are many among my readership who can identify with that phenomenon.

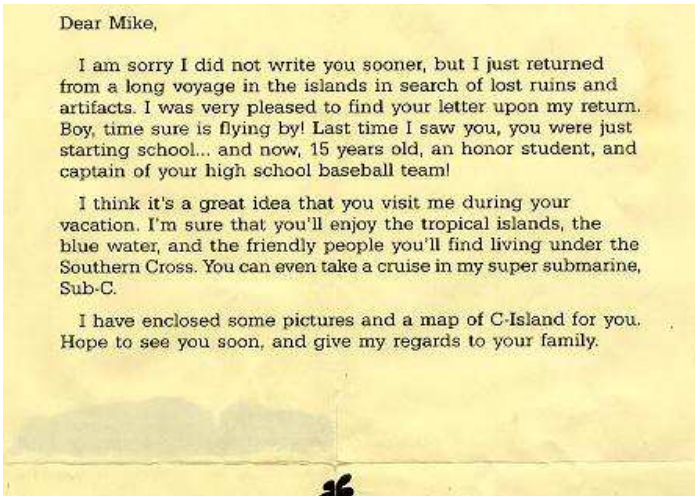
One's success in **Super Mario Bros.** or **Zelda** depends, to some degree, on the player's ability to make those connections. While much of it is intuitive-- as we mentioned before, most people, for example, should be able to figure out how to get to the warp zone in 2-1-- some of it isn't. What I'm talking about, of course, are the maze castles, particularly the castle in 7-4.

The maze in 4-4, while somewhat frustrating, is not insurmountable; a little trial and error and you'll find the solution in no time. The maze in 8-4, while arguably more complex than the others, creates a sense of space with landmarks such as pits, pipes, and enemies.

7-4, by contrast, is extremely abstract. Replaying the game recently, I tried in vain to discover the solution on my lonesome. The time ran out on me and I quickly burned through my pool of lives. I tried to remember the mnemonic device my neighbor had taught me but it had become obscured over the years.

Finally, I took my frustration to the internet. A google search lead me to **themushroomkingdom.net,** which graciously allowed me to reprint their enemy populations chart and thankfully had the solution to the seventh castle maze. And so, Miyamoto's idea worked again: here I was, consulting other gamers to find my answer.

At the same time, it's an idea that shows its age. Just as "topical" movies and novels seem creaky when the audience lacks knowledge of the current events that gave them their context, a video game that depends on the audience having information not present in the game is on shaky ground. Anyone trying to play **StarTropics** without the secret code letter that comes packaged with the instruction manual and contains the code required to progress further in the game knows exactly what I'm talking about.



Dear Mike,

I am sorry I did not write you sooner, but I just returned from a long voyage in the islands in search of lost ruins and artifacts. I was very pleased to find your letter upon my return. Boy, time sure is flying by! Last time I saw you, you were just starting school... and now, 15 years old, an honor student, and captain of your high school baseball team!

I think it's a great idea that you visit me during your vacation. I'm sure that you'll enjoy the tropical islands, the blue water, and the friendly people you'll find living under the Southern Cross. You can even take a cruise in my super submarine, Sub-C.

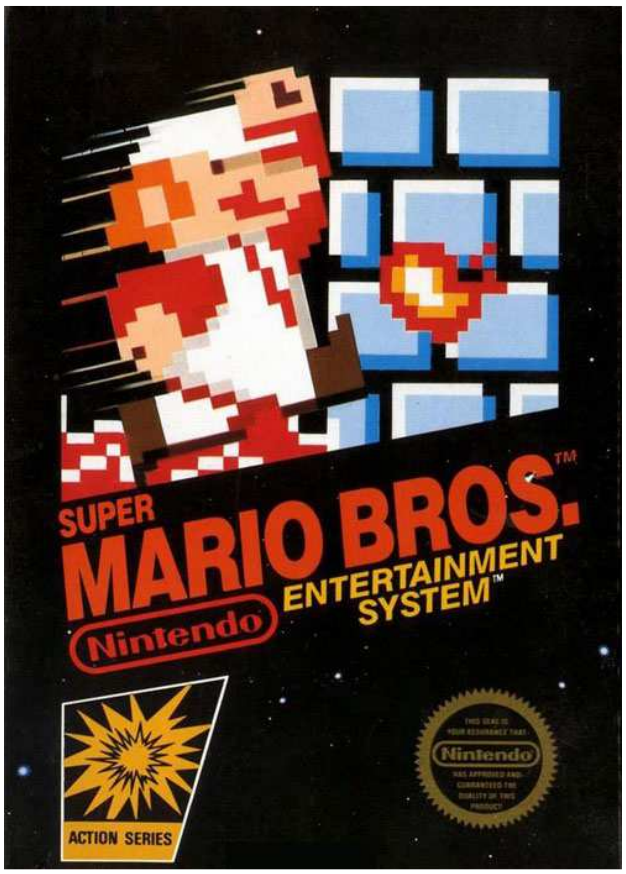
I have enclosed some pictures and a map of C-Island for you. Hope to see you soon, and give my regards to your family.

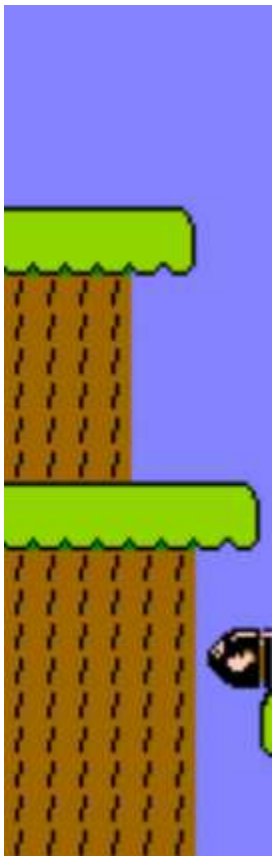
You had to submerge this letter in water in order to reveal the secret code for *StarTropics*. Note the wet spot.

To put it another way: if you were to sit someone who had never played video games before in front of **Super Mario Bros.** and pressed a controller into their hands, chances are they'd get the hang of it. The first level of the game would acquaint them with stomping on enemies, grabbing power-ups, busting bricks, jumping over pits, kicking turtle shells, and looking for secrets. As the game progressed their skill and their knowledge about the game would progress as well. They would pick up on the basics of platforming and, to some degree, of video games in general. But then they'd get to 7-4 and they'd hit a brick wall. The world's most accessible and nearly perfect video game just became inscrutable.

It is a flaw in the game, though it's not the result of an accident or sloppiness; it was a conscious design decision that, for its time, worked. And so I can't necessarily fault the game for it. But I can admit that the time for that sort of social element in gaming has long since passed.

What lesson does this teach today's game designer? It teaches us that while our games should contain secrets, those secrets should be intuitive. The maze in 8-4, with its visual landmarks and clearly-differentiated paths, is much easier to understand and beat than that in 7-4 or even 4-4. There should be clues to help the player along. (Consider, again, the example of the warp zone in 1-2.) Player should be able to figure out those secrets by thinking and learning, not merely by trial-and-error.





Cliff Notes

The problem with writing in-depth game design articles is that the reader is at times overwhelmed by the sheer volume of material. And while I stand by the quality of and necessity for that material, I also understand that some readers are better served by something somewhat more pithy.

I also see the need, by way of conclusion, to quickly summarize the game design concepts that our time with **Super Mario Bros.** have brought to our attention.

And so, to answer both of these needs, here is a collection of suitably pithy maxims and reminders. Those looking for more substance are directed back to the article proper; the following are arranged in the order in which we covered them.

Start with a high concept. Every part of the game should unfold from that center.

More vital than what a power-up does is how the game's power-ups relate to and work with one another.

A game's first level should contain the entire game in miniature.

Introduce a power-up near the thing that will best demonstrate its use. For example, the Super Mushroom is first seen next to some breakable bricks.

Separate each stage into several "mini-stages" to ensure variety, unity, and fairness. Use "rest" areas liberally.

Give the player meaningful choices and meaningful ways to interact with the game world.

Introduce new content as often as possible; make every level count.

Introduce new challenges with a "safety net" in place. Then, as the game progresses, remove that safety net.

In levels themed around speed (such as 2-3 and 4-2), use a more linear level design.

Memorable challenges can be created by inverting the logic behind other challenges. Don't be afraid to turn the game's concept on its ear.

A game's final stage should encapsulate the preceding levels without merely rehashing them.

Some enemies and challenges are memorable because they are *special*. Don't dilute them by reusing them again and again and again. Let some challenges be specific to one and only one level.

No two enemies should be exactly alike. This has less to do with hit points or attack points and more to do with an enemy's vulnerabilities and, most importantly, the way an enemy controls screen space.

Don't throw too many enemies or challenges at the player at the same time. Use mini-stages to space them out and ensure the difficulty is fairly-implemented. Think deeply about how different enemies and obstacles work together to create difficulty.

Challenges that depend on the player doing something in one exact way with no margin for error aren't fair.

Trial and error puzzles are no puzzles at all; secrets need clues and should be intuitive.

Look Closer LOOKING DEEPLY AT GAME DESIGN IN CLASSIC GAMES

The Narshe Incident:

Attack Power as a Source of Meaning in *Final Fantasy VI*

by Tom Russell

Role-playing games, and the **Final Fantasy** series in particular, often depend too much on cinematics, dialogue, and other mimetic elements as sources of thematic, emotional, and intellectual meaning-- often at the expense of the actual game play.[15] But that's not always the case; sometimes, non-mimetic elements-- those exclusive to the art form, and, in this specific case, the genre-- are used in ingenious ways to create these types of meanings in a more visceral and ultimately deeper fashion.

Final Fantasy VI[16] posits a world in turmoil, pitting a rag-tag group of rebels, the Returners, against a powerful, fascistic empire, which seeks to gain control of magic-- a force that decimated the earth long ago and could easily do so again.

The story begins in the small, secluded mining town of Narshe. Imperial forces have heard of a strange and wondrous magical creature-- an Esper-- that was recently found in the mines, frozen in ice. Two imperial soldiers, Vicks and Wedge, are sent to find the esper. Accompanying them is a mysterious girl who can communicate with the esper, under their complete control through the use of a "slave crown".

Now, that's the set-up, and it's all pretty standard stuff, expressed through dialogue and monochromatic cut-scenes. But when the game play begins, the player finds themselves in uncomfortable emotional and intellectual territory, and that discomfort is the result of the game play itself. Let us first trace the sequence of events that we'll be looking at, and then we'll take a look at the ways in which the game play and game design decisions augment and create meaning in this portion of the game.

So, our "heroes"-- the two imperial goons and their brainwashed captive-- storm into the city of Narshe in ultra-powerful MagiTech armour. As they make their way towards the mine, they are attacked by men, dogs, and woolly mammoths-- all of which are destroyed with ease and pose no actual threat to our trio.

They have slightly more trouble with the Whelk, the game's first boss, but soon they find themselves at their objective: the frozen esper. The esper destroys Vicks and Wedge; the girl loses consciousness and wakes later, free from mind-control, in the home of a Returner. When a mob of his fellow Narshe citizens shows up, demanding vengeance for their fallen friends and canines, the girl must try to escape via the mines.

Now alone and deprived of her MagiTech armour, she finds battle with guards and mine monsters to be considerably more difficult. Soon, she is cornered and the floor beneath her gives way. She falls unconscious once more.

---

[15] For a more thorough discussion of this problem, I humbly point the reader towards "The Problem with RPGs", the mammoth cover feature article that makes up the bulk of our second issue.

[16] Initially released in the United States for the Super Nintendo under the title **Final Fantasy III**.

The defenders of Narshe are no match for MagiTech.

Now, of course, the story doesn't end there-- new character Locke and a googol of moogles show up to assist her in her escape.[17] But, for the purposes of our discussion, we'll stop short of that rescue.

If we divide that sequence of game play events into two parts, with the esper encounter serving as the dividing line between the two, it is easy to spot a number of differences: a party of three versus a party of one, MagiTech versus no MagiTech, advancing versus retreating, hunting versus being hunted. In the first part, the player's enemies-- the innocent and hopelessly outgunned defenders of Narshe-- are slaughtered; in the second part, the player's sole character is in danger of being slaughtered herself. The tension and moral ambiguity that result from this sequence of events largely derive *not* from mimetic snippets of dialogue, but from disparities between the attack power of the player and that of the enemies.

It is not uncommon in the first leg of an RPG for the player to control a high-level character in addition to weaker ones. This allows the player to get the hang of turn-based combat, and the specific itineration of that system used in a given game, without having to first contend with a steep difficulty curve. Once the player has been given time to learn the ropes, the high-level character is removed and the game begins to become more difficult. Beyond that, it also gives the player a taste of what's to come later in the game; so much of the genre's power and appeal comes from the gradual accumulation of powers and abilities as the player rackets up experience points, a gradual process that only really pays dividends in the middle or later stages of the game, making the early stages somewhat dull in comparison. By giving the player a peek of what's next, designers are able to maintain their interest early on.

The MagiTech armour effectively acts as that sort of high-level "mentor" character-- it gives the player access to a wider variety of more powerful attacks and smoothes out the difficulty curve. But in this case it's not just a design decision, a way to introduce the player to the system: it's also an aesthetic one.

Vicks and Wedge also act as mentor figures, explaining the ins-and-outs of boss battling in a safe context.

The player is so incredibly powerful in this first half of the Narshe adventure that it's *too* easy. Narshe's defenders don't stand a chance; their attacks are puny against the heavy defense of the MagiTech armour, and not a single one of them can survive one attack. It's really completely unfair, and the ease of the slaughter is stomach-turning.

If the player's party had been less powerful-- still more powerful than its opponents, but allowing the enemy at least the faintest wisp of something resembling a chance-- if, for example, the enemy could survive one or two rounds of fighting, making it more like an actual battle and less like a

[17] Okay, so it's not literally a googol of moogles; it's more like a dozen. But, c'mon, that was fun.

slaughter-- the moral ambiguity would have been greatly diminished.

For if we feel sorry at all for the men and dogs we massacre marching into Narshe, we feel considerable less pity for those men and monsters chasing after our lone heroine as we attempt to flee Narshe by way of the mines. In the second half of this sequence, in which the player is deprived of fellow party members and MagiTech, the disparity between the hit points and attack power of the player and the enemy characters is greatly decreased.

Your progress through that mine is arduous and hard-fought. After each encounter[18], the player heals themselves and tries to subdue their sense of panic. They walk as quickly as they can, hoping they can get just a few more steps before the next random encounter puts them in mortal danger.

Just as loneliness becomes more bitter and unbearable when it has been preceded by love, so too does the difficulty of this second half seem that much more difficult after the ridiculous ease of the first. Feelings of desperation and panic-- one of the few emotions that games can create in a player through non-mimetic means-- abound, and perhaps there is also the sinking, subliminal feeling that you deserve this for murdering those poor defenseless defenders.

A measly 32 points and one party member-- huge difference from 425 and three.

Not only does the change in attack power prompt a deeper emotional and intellectual response from the player, but the way in which that disparity between player and enemy attack power is expressed in an RPG helps to create those deeper meanings.

If, for example, this game was a platformer, and the player was faced with hopelessly underpowered enemies who could be destroyed with one hit, there wouldn't be a sense that it was unfair; it would just be business as usual and the disparity would not be as obvious or stark. But because attack power in an RPG is represented numerically, the disparity is very clearly quantified: the enemy pathetically manages to do one to eight points of damage, barely making a dent into your sixty-plus pool; your attacks, on the other hand, can do over *four-hundred* points of damage.

Such force is blatant overkill-- and that sense of overkill further contributes to the sense of moral discomfort. And the disparity between the player's MagiTech attack power and the player's normal attack power, increases one's feelings of powerlessness.

In this particular example, mimetic and non-mimetic elements work together to create atmosphere, moral ambiguity, and emotional responses. This combination results, I think, in a visceral and thus more meaningful effect than would result from the use of mimetic elements alone; by allowing these meanings to arise from the experience itself and the playing of the game, the designers accomplish what mere words and pictures cannot.

[18] Especially with those %$#&ing hunchback repo men.

Reviews

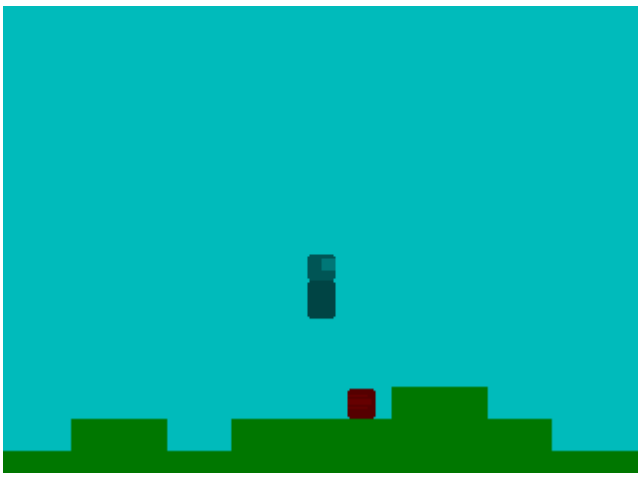
Our reviews section is admittedly a little light this time around. It's a little embarrassing just after winning that **2008 Game Maker Blog Community Award for Best Game Maker Reviews**. To make a long story short, I didn't budget my time properly and it came down to cutting a few reviews or releasing the magazine a month late-- the latter of which would be disastrous, I think, for a quarterly publication, whose grip on the public's mind is already so tenuous.

To make that story slightly longer, it's also a matter of quality. I had several games that I was planning on reviewing-- several games, in fact, that I was excited about reviewing. But I simply haven't played those games long enough and thought deeply about them enough to do them justice. One of those games, some of you might be interested to note, is **Iji**.

I originally planned on reviewing it in our October issue, realized I needed to spend a lot more time with it, and pushed it back to this one. And now here I am, in January, pushing it back again. It's not that I haven't been playing it-- I have-- but that the game requires at least another three months of my time before I can even begin to talk about it. I suppose in some respects that's a review in and of itself.

Anyway; onward! As usual, my open invitation still stands: if you're a game creator who would like to have their game reviewed in these pages-- especially if said game isn't quite getting the attention you feel it deserves-- I'd be more than happy to review it in a future issue. I only review completed games and I will not review fan games. I also do not pull my punches; what you get will be my honest opinion, and you might not like everything I have to say. That's the risk you run; I think the reward, though, is worth it.

Pinball Panda, p. 57



Alternate Path, p. 60

Pinball Panda
Arcade by Barry Atkins
http://www.yoyogames.com/games/show/59739

I'm not a big fan of pachinko, but I am a ginormous fan of Barry Atkins, the gaming academic responsible for *Videogame, Player, Text* and games like **Shush, Daedalus and Son,** and **Triptych**, among others. What I like about Atkins's games is that he's not afraid to take chances; each game is a bold and fearless experiment in game design. Some work better than others, and they might lack the polish and style of a Cactus game, but everything he's done has been interesting.

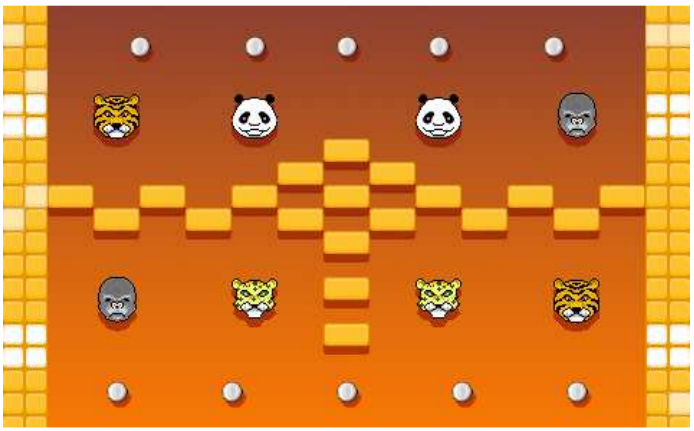
Shush, which I reviewed in the very first issue of the Quarterly and which served as the main point of inspiration for my article "Visionary Sound" that appeared in the eleventh issue of *Game Maker Technology Magazine*, used acoustical elements to quickly impart information that is more difficult to grasp visually. It hinted at some of the artistic possibilities, as yet untapped, regarding the use of sound in video games. **Daedalus and Son** took the basics of a 2-D platformer (platforms, obstacles, treasure, enemies) and turned it on its ear by giving the player a protagonist who soars through the air by way of the mouse. That's right: a mouse-based platformer. **Triptych** is more daring, as it tasks the player with playing three games simultaneously.

These, and every other Atkins game that I've played, spring forth from two of the most important questions in a game designer's repertoire: *what happens if I do this?* and *what else is there?* And these questions are answered with a strong theoretical and practical background. As Paul Eres wrote in "Miyamoto's Framework",

> *If you take Miyamoto's framework and just reject principles at random and replace them with new principles at random, chances are you won't wind up with a new system that works, you'll wind up with a mess. You can't just "reject goals" and "reject the player dying" and make something without them and hope it'll be great, you need to form a new systematic set of principles which work as cohesively together as his do.*

Atkins understands this intuitively; whenever he asks *what happens if* and *what else is there* he ensures that the end result is playable and cohesive. Atkins is one of the most interesting experimentalists working in game design today, and I was pickled pink[19] that one of his games, **Pinball Panda**, has lately garnered the popularity he deserves.

Pinball Panda is, as the reader can no doubt guess from this review's first sentence, a variation on pachinko-style games and, as a consequence, is less experimental and daring than some of his other games. That's not to say that he plays it "straight", as it were, or that he just gives us "another pachinko game"; **Pinball Panda** features a number of bells-and-whistles that not only transform the game play into something that forces me to suspend my usual apathy towards these types of games but also-- and more importantly-- uses the video game art form in important non-mimetic ways. Or, to put that another way: **Pinball Panda** could never exist as a "real" (non-electronic) game at a carnival or arcade, and it's all the better for it.



The game tasks you with "rescuing" pairs of endangered animals from a given board by clonking them on the head with a little ball; as in pachinko, the ball is dropped from the top of the screen and bounces off of various implements on the board. Once the player has aimed and released the ball, he has no control over it; freeing the animals and clearing a board, then, is a matter of understanding innately the angle at which the ball

[19] That's right, pickled; I've spent the last few days in a jar of vinegar.

bounces from a given object if it hits it from this angle or that one.

There is a danger inherent in such a game play mechanic: it's entirely possible that in lesser hands this would become a game of chance and luck rather than skill. I am reminded of one of my own experiments in game design, an overhead action-adventure game in which the player uses bouncing balls to fend off enemies instead of a sword or arrows; since the ball could bounce a different way each time, it was not reliable and thus frustrating.

Atkins heads off this problem at the pass with an exquisitely tight bit of physics programming that ensures that the ball always reacts in the same way under the same circumstances, one hundred percent of the time. If you aim the ball at the same angle from the same position, it will bounce in precisely one way and one way only off of a given object, moving to the next object, et cetera. The game plays completely fair, which is the only way a game like this (or electronic billiards for that matter) could ever actually work.

But did I say it bounced the same way every time? Well, yes and no. Yes, the physics are quite reliable. No, because it's very seldom that the board itself will be the same from one ball to the next. When a pair of animals has been clonked, they'll disappear from the board, which means that a ball sent in their direction will now have to find something else entirely to bounce off of. Some bumpers will also disappear when the ball strikes them, causing you to rethink your trajectory.

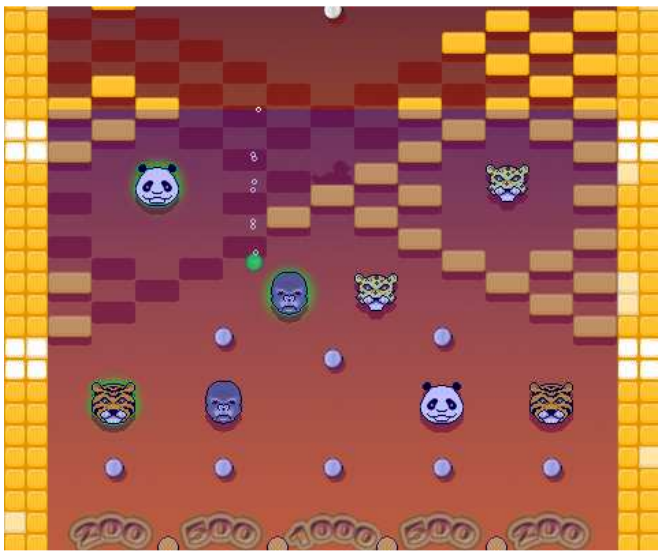
If disappearing bumpers would be difficult but doable to pull off with a pinball or pachinko machine, the target bumpers that adorn some stages would be completely impossible. Striking one of these causes both bumper and ball to disappear and gives the player the equivalent of crosshairs for a cursor. The player can move the crosshairs freely around the board; clicking the mouse button will cause the ball to appear in that spot-- most advantageously, of course, just over the head of a hard-to-reach tiger or panda. This little power-up isn't just a desirable extra but an important and necessary element of the boards on which it appears.

The same goes for the pair of teleporting baskets that make an appearance here-and-there. The ball must fall into the upright basket so that it might drop from the downturned basket. The boards in which it appears are (naturally) comprised of two separate sections with the animals necessary to clear the board scattered in both.

Atkins's tweaks and twists don't merely make it easier; some of them make the game harder. When was the last time you played a pinball or pachinko game that featured *enemies*? (No, **Sonic Spinball** doesn't count.) **Pinball Panda** pits players against poltergeists, ready and will to eat your balls. But they're not nearly as pesky as the horizontally-moving platform that bounces the balls upwards, thus impeding your downward progress, or the rubber ducky.

If your ball lights upon yonder synthetically feathered friend of Ernie, the board will quickly fill up with water. And if your ball touches said water, it loses all speed and trajectory and bounciness and just sinks, straight-down and without mercy. If you hit the rubber ducky, especially if you do so early

on in the boards in which he appears, your chances of clearing the board in the prescribed seven balls are very slim indeed.



That was a cause of mild frustration here at chez Russell. I had no choice but to eject one meaningless ball after another, using up my remaining six so that I could start the board over again. If, in future releases, Mr. Atkins would be so kind as to add a "suicide" button, it would make those rubber ducky stages less Sisyphean.

A level select screen-- something I've become accustomed to in his previous releases, such as **Daedalus and Son** and **Dual Core**-- would also be a welcome addition and I'm frankly surprised that he did not include such a feature in this game. While **Pinball Panda** is appealingly short-- by my recollection it's just over a dozen or so boards and thus never overstays its welcome-- because some of the boards (i.e., the rubber ducky stages and one of the basket levels) are a bit frustrating one dreads having to replay them.

Still, Atkins offers us the next best thing; upon using up all seven balls, the player is given some options. He can buy "just one more ball", at the cost of 1,500 points, restart the level for 5,000, or skip it entirely for 7,500. These options give less skillful players some wiggle room, making it more accessible. It's a remarkably smart game design decision.

And this game is full of them. If this had been just another pachinko game, a simulation, I probably wouldn't have bothered reviewing it. I'm going to put on my philistine hat and say that I just plain don't understand the point of a video game that slavishly recreates another type of game, such as billiards, golf, or (God help us all) a die-rolling board game. When I play a video game, I expect it to provide an experience that only a video game can provide.

And that is what Atkins provides with **Pinball Panda**: the target bumpers, the disappearing animal pairs, the baskets, the ghosts, and even that damnable rubber ducky-- all these elements make glorious use of the art form itself. They make the game interesting for me as a writer and they make the game fun for me as a somewhat persnickety player. And they do it in a way that works within the basic framework that pachinko provides, instead of working against it (like the aforementioned **Sonic Spinball**).

And so, even in this, perhaps the least daring of the Barry Atkins games I've played, he's still experimenting, still asking those all-important and innovative questions of *what else is there* and *what happens if I do this instead*; most importantly of all, the answers he finds still result in an eminently playable and enjoyable game.

Alternate Path

Puzzle Platformer by David Yates

http://www.yoyogames.com/games/show/50097

Alternate Path reminds me a bit of the game **Lost in the Desert**, which, as canny readers are no doubt aware, was reviewed in the fourth issue of the Quarterly. In fact, I was seriously tempted to reuse the genre descriptor I had settled on for Sam R-B's minimalist masterwork of a game, *avant-garde experience*, in order to describe this effective and startling little game by David Yates, for in both cases the respective creators are interested in exploring, expanding, and experimenting with the possibilities inherent in the video game art form. Both are unusual, interesting, and important works of art from a purely formal perspective that are probably slightly lacking in terms of deep meaning.

But while that classification, avant-garde experience, could be applied to the game currently under consideration, the eagle-eyed reader will note that I went with the far more common *puzzle platformer*. This is because I find the differences between the two games to be far more striking than the similarities. Both might be experiments in form, but they are experiments of a wholly different nature: **Lost in the Desert** stretches the definition of "game" to its breaking point while still fully embracing the art form itself; **Alternate Path**-- uh, alternatively-- instead asks what a "game" is by playing with the conventions and grammar of video games in general and of the framework created by Shigeru Miyamoto in particular.[20]

There is, however, one more similarity between **Desert** and **Path** that I will remark upon before enumerating and elucidating upon that aforementioned difference: just as I did with last issue's review of **Lost in the Desert**, I am strongly advising that you play **Alternate Path** *before* reading any further. In both cases, the balance of the review seeks to discuss the game in detail and in depth, spoiling every secret and investigating every nook and cranny. A full, complete, and mature understanding of the game cannot be conveyed if I spend the next few pages talking around it and hinting at things that will only be clear in hindsight. So please, dear reader, go and play the game already.

I trust if you're still reading that you've done just that, and now we can get to the business of discussing it. There are a number of ways of looking at **Alternate Path**: as a collection of puzzles, as an experiment, as a meditation on gaming, as a sort of history of platformers. And I'm sure there are plenty of others that I'm missing but I'll start with that last one.

The first "level" of the game allows the player to move and to jump using the arrow keys; the only way to dispose of the enemies in this scarcely-populated area is to jump on their heads, *ala* **Super Mario Bros.** One touch from any of these enemies will kill us, thus sending us back to the very first level. This is platforming at its most rudimentary.

But as the game progresses, that formula is complicated and transformed. We gain a gun that can destroy barriers-- shades of **Mega Man 2** and **Blaster Master**, not to mention countless others. We gain a health bar and lose the ability to jump on enemies, instead utilizing our gun-- now the standard for action games. We gain a second weapon and the ability to switch between them instantaneously with the touch of a single button; today's greatest platformers, such as the **Ratchet and Clank** series[21], make weapon-switching a central feature of the game play.

We've gone from simple to complex, from light-and-bouncy (or, given the game's decidedly low-fi visual style, what passes for it) to dark and murky as grays and blacks eventually give way to brooding reds. Now, I'm certainly not trying to read too much into this here-- I don't think, for example,

[20] Not for the first time, I direct the reader to Paul Eres's terrific essay *Miyamoto's Framework* in this very same issue.

[21] "Ah," you say, "but what about this *systemic cohesion* you blather on and on about, issue after issue? Doesn't that series throw it to the wind?" Yes, it does, and I feel it is flawed for that very same reason. It's very irritating when one is playing a platformer to be unable to advance because one is bad at flight simulators and turret-shooters.

that it represents, say, a lament for the platformer's loss of innocence or some such nonsense-- but I found this meta-narrative, this Short History of Platformers in five minutes or less, to be emblematic of the game's approach, which is largely referential and metatextual.

Or, to put it another way: structuring it as a history of the platformer makes sense because it foregrounds the fact that this is a game about gaming. Take, for example, the two major puzzles that the game confronts us with.

First, there is the giant pit puzzle. As we enter the room, we are helpfully told to press the F5 function key to save. To our right, there is a large pit, too large for us to cross over even with our recent jump power-up. There are no invisible blocks; we try to jump across and we fall into the pit. The room restarts and we're told again: press F5 to save. We just might do that, so we can try to solve this puzzle later.

But the solution is staring us in the face: *press F5 to save*. We jump towards the abyss and press the F5 button and we are "saved"; we arrive on the other side of the ledge.

It's such a smart little puzzle, calling into question part of video gaming grammar that we had taken for granted: the meaning of the word "save". It's an intuitive puzzle-- it's not *that* hard to figure out, all the clues we need are present-- but it doesn't beat us over the head either. It requires us to think.

Not only does it require us to think in order to solve the puzzle, it requires us to think about the puzzle. About the meaning of "save". In my case, it reminded me of moments in my life when I had made a boneheaded mistake and wished I could do it over, that I could just reload from a previous point in my day. It got me thinking about how dangerous a place like the Mushroom Kingdom would really be-- how many lives would be lost each year as the result of those bottomless pits and wayward turtle shells.

And that got me thinking about how this game doesn't feature a lives system, and how a lot of modern action games have eschewed that convention as well. And I reflected on the game's meta-narrative, and I wondered how much richer it might be if the player started the game with a finite number of lives that gave way to the unlimited-lives health-bar system. It got me thinking about other ways the meta-narrative could be complicated and perhaps improved.

And then I reflected on how that really wouldn't be fair to a player; this pit puzzle, taking place before the acquisition of the health bar, would be brutal if the player had a finite number of lives. But it would still be a cool idea, to do a meta-narrative game, a platformer that encapsulates the history of the genre, and I started thinking about ways to implement such an idea.

All this thinking-- thinking about my life, about my gaming experience, about game design: it all sprung out of this little puzzle. And I'm not trying to argue that Mr. Yates intended me to think along those lines when he created this puzzle; I'm saying that he intended us to think, period, to reflect and meditate upon games and language and the language of games.

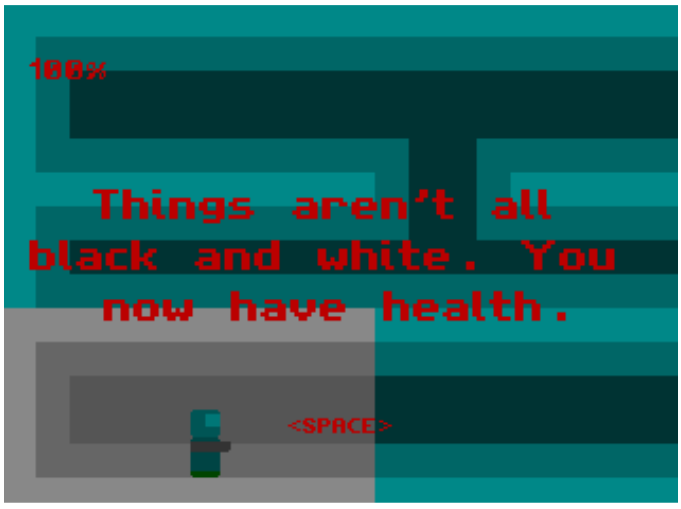
The next major puzzle, and the one that's probably the most likely to get people stuck, is the "What do you need?" segment. We're faced with a door or barrier and we find an rectangular object lying on the floor in the same room. "This is exactly what you need," we're told. But what is it that we need? We're prompted to type in the object's name.

I got stuck on this puzzle myself, typing "key" and "pass" and other variants. And while those are common barrier-destroyers in video games, it was just as common, back in the day, to use a gun.

And that got me thinking about the ludicrousness of that trope. If I knew someone who used a gun whenever confronted with a door or barrier, I'd frankly be a little frightened of them. But it makes perfect sense in the context of gaming, as part of the language of games.

There a lot of things like that: the idiosyncratic way gamers use "save", for example. The presence of extra lives and checkpoints. Boss battles. Health bars: why can we function just as well at 10% health as we can at 100? The reliance on violence, even cartoony violence, to solve conflicts and achieve goals.

And all of it, like Miyamoto's framework, is pretty arbitrary. Sure, it all works, it all makes sense, and it's all grown organically. But why are those tropes the ones that are dominant? Why is that the language of games? *What else is there?*



What's more, this language has nothing to do with life. All other art forms have something to do with life, or else it's just pretty nonsense. It's a discomfiting bit of introspection to go through as a proponent of games-as-art.

And, again, I'm not saying that Yates necessarily meant us to think that deeply about that specifically, but his puzzles are constructed in such a way that they ask us to think deeply about something. About games and the way they are constructed.

This comes to a head at the game's conclusion, in which we're confronted with the creature responsible for our alternate path: the AntiPlayer. His purpose, he tells us, is to stop us from achieving our purpose; our purpose "was to play this game." But, one interesting-if-slightly-overlong boss battle later, someone (the game's creator?) explains that the goal of the game was to defeat the AntiPlayer, and having done that, we have completed the game. "Makes one wonder just how alternate the AntiPlayer's path for your was..."

Granted, this is a bit of story, of dialogue; it's a mimetic thing. At the same time, the game play itself builds to this question. The puzzles are built around it, around the act of asking questions and thinking about the art of gaming.

Alternate Path is an interesting experiment, greater both in ambition and achievement than the last game of Yates's that I reviewed, **Elements of Escape.** It investigates the way in which meaning is communicated in video games, and asks why deeper meaning-- the stuff of life-- isn't communicated through those non-mimetic means.

It's an important and unsettling question. It does not provide us with an easy or pat answer. Maybe that's because games can't tackle the deep meaning of real art on their own.

At the same time, I'd like to think that they can, and the experimental game that pulls it off-- *that* will be something to see, that will be the holy grail. And by demonstrating his capacity to ask these uncomfortable questions, I think Yates is closer to finding that grail than most of us.

I eagerly look forward to his next experiment with the hope that it's less Godard and more Bresson. Good luck and godspeed.

Gaming with One Hand
by Peter Jurich

Years ago, I vanquished a giant lizard and saved a talking mushroom only to discover that the princess I came for was in another castle. I was discouraged, naturally, but I pressed on regardless. The second castle bore the same misgiving, and so I tried the third. The princess must be here! I told myself. Loquacious fungi are a rare phenomenon and surely it was a stroke of luck to even find two!

(Against all better judgment, I did not report my findings to biologists, for who in their right mind would believe a four-year-old boy?)

"Play with both hands," my mother would admonish me from the kitchen threshold. And for a few brief seconds, I'd obey, holding the controller properly between the palms of two hands, the way Shigeru Miyamoto had intended. The poor woman considered herself lucky that her son was even alive. Seeing him neglect his therapy would break her dear old heart.

And so after she left to tend to whatever pressing matters in the kitchen purloin the attention of mothers, I was back to playing **Super Mario Bros.** the way in which I felt most comfortable: laying on my belly, propping myself up with my left arm, controller laid in front of me, my right hand stretched across its breadth.

Nintendo Entertainment Systems and I were not a match made in Heaven. In fact, a Nintendo was purchased for me not for the entertainment it provided, but to strengthen my left hand. You see, childhood brain trauma had left my circuitry awry and resulted in an inoperable left hand. At first, I lacked full gross motor skills—my hand would float up in the air unless I consciously pulled it down—but I eventually regained normalcy. However, I never quite regained my fine motor skills, which makes difficult any exercise involving the strategic pushing of tiny buttons on a small controller (i.e. Nintendo).

It was never in my itinerary to disobey my mother, but be it as it may, it was never in Princess Toadstool's itinerary to be kidnapped by Bowser. My negligence was fully justified, I felt, by my glimmering display of heroics. Nintendo practically begged me to play with one hand! There were only five buttons to the controller, and, with the Select button rarely in use, I was already a step ahead of the game!

Through the years, I have traversed across many different worlds and seen even more foreign creatures, from the helpful space monkeys in the depths of Planet Zeebs to the rabbites of Mana. I have conquered the Moldorm guarding Hera's Tower and thwarted the Hamburglar in McDonald Land. I have defeated dozens of Dr. Wily's robot creations and I took some time afterwards to shoot Agent 006 off of the Antenna. I have done these things one-handedly.

These milestones were not easy, however. I have always approached any new gaming vehicle with an apprehension understood only by those maybe missing a limb. (Think amputee playing **Dance Dance Revolution**.) When Super Nintendo came out, I was shocked at the addition of four extra buttons, but Nintendo had trained me for that. Where my fine motor skills lacked in one hand, they far exceeded in the other. It was a quick

transition. Only one more button each—X and Y—for my pinky and ring fingers, and who really needs to jump with L and R in **Mario Kart** anyway? It was just a hiccup in the design. The infallible and ever-humanitarian design of Nintendo Entertainment. Who could possibly begrudge those responsible for the happy childhoods of young'ns everywhere. After all, nobody's perfect… except Squaresoft.

Super Nintendo was also my first introduction to RPGs and thus my emergence into manhood. I hold Squaresoft—a company renown for games of that genre—responsible for most of my childhood. **Final Fantasy III**[22] was a door to a different world: Sid was the father I never had; Terra was the girl I never talked to; defeating baddies was the sport I never played; and chocobos were the chickens I never fostered.

RPGs also proved to me a serious point of discussion that which proved to me that not every video game needs to be a race between time and your heart rate. These magic cartridges of hope were driven not by game play, but by dialogue. Games with stories! What an idea! Long gone was the idea that I should open up to a new game, controlling an indiscernible jumble of pixels and be expected to care for it like it was my own, like it mattered that I jumped over some bottomless pit. No sir; Squaresoft games had emotion!

What struck me most about RPGs—maybe the very reason I enjoyed them so much—was they were stress free in physicality. Very rare was a time when I had to make a quick decision that required the use of both hands collaborating on in a beautiful panoply of two-hand teamwork. Even in the heat of notorious battle was I eased at the thought that I could take a nap and neither monster nor fighter would make a bold decision independent of myself.

Super Nintendo and I would be together forever. **Secret of Mana** followed and one-upped only by **Chrono Trigger**, amazing games the two of them! I had finally found my home in the world of videogames.

When Nintendo 64 came out, the company had made their message abundantly clear: we do not support the one-handed community. I was ebullient of the system's coming to be in my household, but discouraged again when the controller presented itself as a sort of three-pronged element fit for giants, or at least ambidextrous people. From here on out, I had recognized Nintendo no longer as a friend, but as a blood-thirsty enemy. I failed to heed the foreshadowing omens bestowed on me from the four extra buttons that Super Nintendo betrayed, but this was far worse. This was a battle cry from a beast with four C-buttons for eyes and a control stick as a tongue.

Biting the bullet was not an option. After all, I had pestered my mother brutally into driving me to Target to get the damn machine. And so seven lives and a bucketful of tears into **Super Mario 64**, I manned up. I had decided it was my mission—my prerogative—to foil Nintendo's dastardly plot and incorporate with N64 the same brand of gaming that had kept me so solitarily happy for years. If I

[22] Editor's Note: He means **Final Fantasy VI,** which was originally released state-side as **Final Fantasy III.**

wanted to survive, I'd have to fight. Avian flu epidemics are not as ambitious as I was.

And over the years, I was rewarded with some of the finest compliments anyone could ever have asked for: "You know Peter," said my friend Chris, "it really pisses me off that you play with one hand and you can still kick all of our asses." Chris was referring to **GoldenEye 007**, debatably the most successful videogame on N64. Friendships were created over **GoldenEye**. Awkward play dates among the children of courting parents turned into nights of mutual respect.

Most actors masturbate to their break-out film; Pierce Brosnan plays **GoldenEye**.

I was at the top of my game. Not only was I an unbeatable force, but that was coupled with the personal victory of rising above the improbable. I was a hit at parties. Looking back, I realize that I may have been viewed more as a circus freak—"Come see The Boy Who Plays Nintendo 64 with One Hand!"—but I relished in it at the time, impressing those more socially superior than I. Those were the good years.

Of course, they did not last forever. My new adversary was released and Nintendo 64 became a thing of the past. Polygons were no longer cool, I guess. GameCube was released and I was challenged again. This time, I would not come out on top. I had won several battles against Nintendo Entertainment, but it was time to raise that white flag. For anyone with access to a GameCube controller, try playing on it with one hand. It cannot be done without supererogatory use of the pause button as you switch from one position to the next. I tried profusely, but was generally met with failure. The logical route for me in my devastation was that I should give up gaming for good.

Metroid Prime looked amazing. Salt in the wound.

Nintendo Wii came out and I wasn't even fazed. Didn't care. Nintendo certainly wanted to rub it in my face. Oh sure, some games required only one controller, which was actually meant to be played with one hand, but the better games utilized the new technology to its full capacity. "You can't play the new **Zelda**," the teased, "but the new **Monkeyball** is pretty good, too!" I was left by the wayside. My best friend had been taken away from me. I spent some time in a dark place; I had to reintegrate myself back into society. Saturday nights were no longer devoted to exploring new adventures with nothing but a broadsword and a bag of Combos. I had to be social. I had to start making friends. I had to talk to people.

I guess that's why I became a writer.

But I digress. Currently, I am residing in January of 2009, and it has been years—2004 maybe—since I have been addicted to a videogame. I have not felt the bliss of addiction, nor the warm touch of an overheated system in that long. It was been lonely.

* * *

Yesterday afternoon, I sat down to start pumping out this article. It was ten or so days passed the deadline, but better late then never, right? Yesterday, I wrote the part about **Final Fantasy III** (I'll be criticized by pundits for referring to it as "III"[23]) and I felt reminiscent. I decided to scourge my old stomping ground, IGN.com, and check up on the latest in gaming. That's when it occurred to me: There's a new Final Fantasy out and I haven't played it? That's unheard of!

And so on that same day, I decided to take a break from writing to treat myself to a few hours as the eighth grader I once was. I went to GameStop to purchase **Final Fantast XII**. My face must've looked belligerent upon entering. So many things I've missed out on. The store clerks gaped at me as I strode passed the Xbox, Wii, Game Boy DS, Playstation 3 displays, into the dusty and dank

[23] And editors.

corner where they had Playstation 2 games. I did not find **Final Fantasy XII** and therefore inquired with the clerk.

He looked concerned at me. "I'm sorry. We don't have any used copies. Is new OK?"

"That's fine," I said. I got out my checkbook and fully prepared to drop fifty bucks.

Playstation 2 games are, after all, great.

"That'll be thirteen dollars," the clerk said.

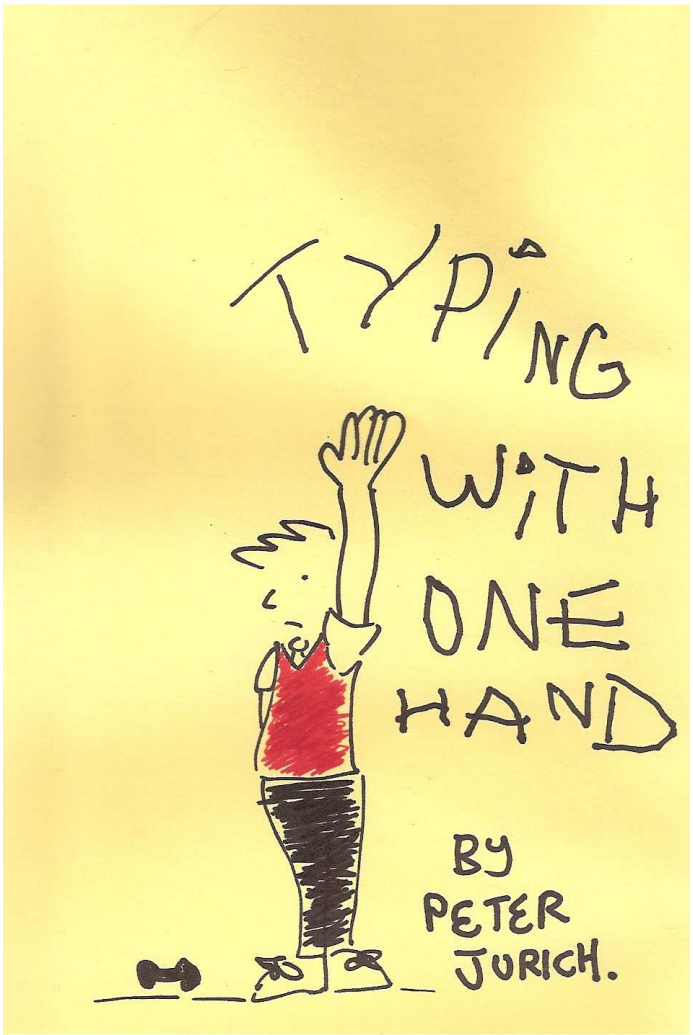
Thirteen dollars! Is that what has become of my childhood? Squaresoft is now Square Enix and the world is upside down. Apparently, Playstation 2 is no longer the leading form of gaming experience. Not if the most anticipated game of 2006 was only thirteen dollars. I can only imagine that that same clerk snickered when I left, exalting his own cleverness. "I sure fooled him," he said. "Thirteen bucks, heh heh heh. For a dinosaur!"

Strange things now. I got home and hooked up my PS2, inviting a very old friend over to discuss our distant falling out. We sat uncomfortably for awhile, my PS2 and I. I offered it something to eat, but it declined. I can't blame it; we were both still hurting.

I pulled up a chair in front of my television and was transported back to a different lifetime. It was a Saturday night and my mother was making chocolate chip cookies, just like the old days. I began playing and, on a whim, I decided to try playing with both hands…

Supposedly, my left hand had become less incapable over the many years I avoided videogames because I was amazed at how nimble my thumb had become! I was using a control stick and ably moving in circles! What a relief! What an amazing assortment of properly functioning brain synapses! I have been delivered from evil!

And so that is my current endeavor. I am playing the newest **Final Fantasy** installment and my faith is restored in gaming. I do not quite know what the future holds for me, but I do know I've rightly enjoyed having a well-to-do bank account over the years and do not look forward to the day when I splurge on something ghastly that will purloin my time. I think my future is hopeful, however, and although gaming with one hand will always be a part of me, it is only a matter of time before I take the necessary steps that will provide me a comfortable existence as a happy recluse, alone with my Nintendo.



Editor's Note: Peter's piece admittedly revolves more around console gaming, but I feel his story raises important points for designers of computer games as well. Control schemes should stress not only simplicity but accessibility. While not every game can be played with one hand, I think we can all agree that we should do all we can to ensure our games are inclusive instead of exclusive.

Next Time

Casual Friday

Correspondence and submissions can be sent to **russellsquarterly@yahoo.com**. Please use one of the following subject headings: Letters, Article Submission, Review Request, or Fiction Submission. Our articles focus on game design, both specific and in abstract. We review only freeware games; we do not review works-in-progress or "fan games"; send us a download link, not an attachment. Fiction submissions should be centered around the video game art form. We do not accept fan fiction of any sort; no one wants to read your Banjo Kazooie hentai.

You retain all rights to any articles or fiction, including the right to reprint and resubmit it to other websites or publishers; you understand that we are an online magazine and that your work printed here-in will always be freely available and cannot be removed from an issue once it is released.

Thank you for reading, commenting on, and supporting this magazine.

I'm the strongest woman in the world.